Biblioteka Główna w Gliwicach
P1877/96

Jak płacić programistom?

miesięcznik profesjonalistów
9
1996

informatyka

Cena: 5 zł. (50 000 zł.) ISSN 0542 9951 WYDAWNICTWO SIGMA NOT

Pismo informatyczne - ukazuje się od 1965 roku

Aplikacja zdrowego rozsądku

wywiad z dr. Markiem Gondzio - prezesem Stowarzyszenia PRO

Relacyjne czy obiektowe?

System informacyjny jako przedsięwzięcie

Najstarsze ale nowoczesne

Witam Czytelników *Informatyki*. W trzecim numerze wydawanym przez nowy zespół redakcyjny chciałbym zaproponować Państwu nową formułę pisma.

Atutem nowej Redakcji, który zawdzięczamy naszym poprzednikom, jest to, że przejęliśmy po nich poważne krajowe czasopismo dla specjalistów, mające ugruntowaną tradycję i renomę. Jest to ogromny i cenny kapitał, który postaramy się nie tylko zachować, ale powiększać. W rozmowach z czytelnikami *Informatyki*, do których należą prawie wszyscy absolwenci wydziałów elektroniki i informatyki, padały uwagi, że pismo ma nieatrakcyjną szatę graficzną, zbyt mało jest w nim ciekawych materiałów oraz, że jest przeznaczone dla wąskiego grona czytelników. No, to nie jest jeszcze tak źle - mogłoby być przecież oceniane jako słabe merytorycznie czy nieobiektywne, a takie wady usuwa się znacznie trudniej.

Wielu moich rozmówców wskazywało na brak krajowego czasopisma informatycznego - jak to określali - *dla dorosłych*. Z drugiej strony wydawanie stricte naukowego czasopisma z dziedziny technicznej w języku polskim, jak to było dotychczas, nie wydaje się uzasadnione. W związku z tym zasadnicza zmiana formuły pisma polega na tym, że choć nadal będzie ono tworzone przez środowiska naukowe, to nie będzie już przeznaczone wyłącznie dla pracowników nauki, ale także dla praktyków. Im właśnie bardzo potrzebne jest źródło obiektywnej i pogłębionej wiedzy o najnowszych rozwiązaniach i zastosowaniach technik informatycznych.

Chcemy, aby *Informatyka* stała się pismem czytanym przez użytkowników, producentów i dostawców produktów i rozwiązań z zakresu technik komputerowych z przedsiębiorstw przemysłowych, firm komputerowych, administracji rządowej i samorządowej. Naukowcy i studenci znajdą zaś w *Informatyce* miejsce na swoje wypowiedzi, a także informacje o konferencjach i zjazdach naukowych, zagadnieniach kształcenia informatyków oraz zastosowaniach technik komputerowych w różnych dziedzinach nauki. *Informatyka* pozostanie więc umiejscowiona w środowisku naukowym, nie aspirując jednak do rangi pisma naukowego.

Takie założenie wiąże się z odejściem od formy artykułów typowej dla prac naukowych, a więc specyficznego języka, licznych odniesień do literatury oraz skomplikowanych wzorów, diagramów i tabel. W doborze materiałów do druku będziemy preferowali te z nich, które uwypuklają stronę praktyczną i ilościową omawianych zagadnień.

Wiadomo, że wiedzę zdobytą na uczelni trzeba ciągle uzupełniać, gdyż zwłaszcza w informatyce szybko się ona starzeje. Można ją zdobywać czytając czasopisma zagraniczne. Ich poważną wadą jest brak odniesień i nieprzystawalność wielu omawianych w nich problemów do polskiej specyfiki. Wbrew pozorom nie zawsze zagadnienia techniczne mają charakter globalny. Dlatego też pamiętając, że ukazujemy się na polskim rynku prasowym, będziemy szczególnie zainteresowani problematyką związaną z informatyką w polskiej gospodarce, nauce i edukacji.

Informatyka nie jest już dziedziną, której zakres opisywały tylko dwa słowa - komputer i oprogramowanie. Zaczęła przenikać się wzajemnie z bardzo dynamicznie rozwijającymi się ostatnio telekomunikacją i technikami zarządzania. Ta tendencja zostanie uwzględniona także przez nas. Na początek stworzyliśmy nowy dział, prowadzony przez Jerzego Szyllera, poświęcony metodyce projektowania i wdrażania oraz eksploatacji systemów informatycznych w przemyśle, instytucjach finansowych i urzędach administracji. Planujemy uczynić ten dział naszym *okrętem flagowym*.

Materiały zamieszczane w *Informatyce* będą tak jak dotychczas prezentować najnowsze rozwiązania oraz światowe trendy rozwoju informatyki w zakresie technologii sprzętowych i oprogramowania oraz ich zastosowań. Wprowadzony w tym celu przez Krystynę Karwicką-Rychlewicz dział wiadomości o wydarzeniach związanych z informatyką będzie nadal stanowił ważny składnik naszego pisma.

Informatyka będzie także forum dla dyskusji o problemach zawodowych informatyków, rynku pracy oraz o podnoszeniu kwalifikacji. Wśród innych ważnych zagadnień, które chcemy poruszać w miesięczniku są: standardy, terminologia specjalistyczna, prawo autorskie, internet, bezpieczeństwo danych oraz ekonomiczne aspekty wdrażania informatyki. Każdy numer miesięcznika będzie miał tzw. temat wiodący czyli główny artykuł lub serię artykułów prezentujących wybrany temat.

Ostatnio powstało kilka nowych organizacji i stowarzyszeń zrzeszających użytkowników i dostawców produktów związanych z informatyką. Liczymy na to, że wszystkie one - stare i nowe - będą z nami współpracować podobnie jak dotychczas czyniło to Polskie Towarzystwo Informatyczne. Nasze pismo będzie miejscem, gdzie można znaleźć informacje o ich działalności.

Zamierzamy także organizować spotkania klubu *Informatyki*, na których byłby omawiany określony temat, a przebieg dyskusji będziemy przedstawiać na łamach pisma. Oczekujemy na propozycje tematyki takich spotkań i ewentualnie pomoc w ich organizacji.

Choć naszym celem nie jest stworzenie pisma do przeglądania, lecz do czytania, to wymogi współczesnego rynku i własne ambicje estetyczne skłaniają nas do zaproponowania bardziej atrakcyjnej formy edytorskiej miesięcznika. Będziemy więc ją stopniowo zmieniać, ale ponieważ jest to przedsięwzięcie kosztowne i niełatwe, prosimy o wyrozumiałość dla niedociągnięć w początkowym okresie tego procesu.

Informatyka nie stanie się pismem czysto komercyjnym, lecz trwanie w przyjętej wiele lat temu formule nie znajduje już u czytelników odzewu, mierzonego liczbą sprzedanych egzemplarzy. Z drugiej strony nadając pismu bardziej rynkowy charakter, nie oczekujemy masowego nakładu, gdyż nadal nasz miesięcznik będzie przeznaczony dla elity, która z natury rzeczy nie może być liczna.

Życząc miłej i pożytecznej lektury, zwracam się do Państwa z propozycją współuczestniczenia w kształtowaniu naszego pisma jako jego Autorzy i życzliwi, mam nadzieję, krytycy.

Redaktor Naczelny

Dziękuję dr. Wacławowi Iszkowskiemu, prezesowi Polskiej Izby Informatyki i Telekomunikacji, niegdyś także redaktorowi *Informatyki*, za twórcze dyskusje i cenne uwagi dotyczące formuły naszego pisma.

informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124

Uwaga
ukazuje się już elektroniczna wersja naszego miesięcznika. Oto nasz adres w Internecie:
http://www.pol.pl/informatyka/

nr 9 1996
Wrzesień
Rok wydania XXXII

Biblioteka Główna Politechniki Śląskiej w Gliwicach
P 1877/96

REDAGUJE ZESPÓŁ:
dr Lesław **WAWRZONEK**
(redaktor naczelny)
mgr Krystyna
KARWICKA-RYCHLEWICZ
(redaktor działu)
mgr Jerzy **SZYLLER**
(redaktor)
mgr Zdzisław **ŻURAKOWSKI**
(redaktor działu)
Agnieszka **PAPIS**
(sekretarz redakcji)
Alina **KLEPACZ**
(administracja redakcji)

KOLEGIUM REDAKCYJNE:
prof. dr hab. Leonard BOLC,
mgr inż. Piotr FUGLEWICZ,
prof. dr hab. Jan GOLIŃSKI,
dr inż. Zenon KULPA,
prof. dr inż. Jan MULAWKA,
prof. dr hab. Wojciech OLEJNICZAK,
mgr inż. Jan RYŻKO,
dr Witold STANISZKIS,
dr inż. Jacek STOCHLAK,
prof. dr hab. Maciej STOLARSKI,
prof. dr hab. Zdzisław SZYJEWSKI,
prof. dr hab. inż.
Ryszard TADEUSIEWICZ,
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004
Redakcja:
00-950 Warszawa,
ul. Ratuszowa 11, p. 628 i 644
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail:
informat@pol.pl

Materiałów nie zamówionych redakcja nie zwraca

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach $3^1/_2$" – w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6 – oraz odbitki na papierze. Teksty nie mogą przekraczać 20 tys. znaków.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

Ogłoszenia przyjmują:
- **Redakcja**
- **Dział Reklamy i Marketingu**
  00-950 Warszawa,
  ul. Mazowiecka 12
  tel.: 27-43-66,
  fax: 26-80-16

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 233/96

W numerze :

Informacje 2-5

Wywiad

Aplikacja zdrowego rozsądku 6-9
wywiad z dr. Markiem Gondzio,
prezesem Stowarzyszenia Polski Rynek Oprogramowania

Systemy informacyjne

INFORMATYKA w dużych projektach informacyjnych 10-11
Jerzy Szyller

System informacyjny jako przedsięwzięcie 11-14
Piotr Sobolewski

Publikacje

Wykorzystanie strukturalnych technik CASE w analizie systemów informacyjnych 15-16
Paweł Wiechecki, Piotr Kuboszek

Relacyjne a obiektowe bazy danych - rywalizacja czy koegzystencja? 17-21
Witold Abramowicz, Grzegorz Bartoszewicz, Leszek A. Maciaszek

Jak płacić programistom? 22-26
Grażyna Hołodnik-Janczura

Systemy temporalnych baz danych 27-31
Krzysztof Kania, Jerzy Gołuchowski

Terminologia

***Reengineering* - a po polsku?** 32
Jerzy Szyller

informacje

InfoMan'96

V Targi Zarządzania Informacją InfoMan'96 – Administracja i Bankowość odbędą się w Gdańsku w dniach 22–25 października 1996 r. Patronat nad imprezą sprawują: Departament Informatyki URM, Związek Banków Polskich i Polska Izba Informatyki i Telekomunikacji. Organizatorami są: Gdańska Akademia Bankowa i Ośrodek Informatyki TBD Urzędu Wojewódzkiego w Gdańsku. Na targach będą zaprezentowane systemy i oprogramowanie dla administracji, bankowości, rynku kapitałowego i inwestycyjnego, ubezpieczeń i telekomunikacji oraz narzędzia do tworzenia relacyjnych baz danych. Do szerzej dyskutowanych tematów należą archiwizacja i wizualizacja danych oraz ich transmisja i ochrona. Wykłady zostaną pogrupowane w kilku blokach seminaryjnych, m.in.: administracja, bankowość, wspomaganie zarządzania finansowego, telekomunikacja, ubezpieczenia. W czasie targów przyznana będzie nagroda im. Stanisława Staszica. W sprawach organizacyjnych należy kontaktować się z: Międzynarodowe Targi Gdańskie SA, ul. Beniowskiego 5, 80-382 Gdańsk, tel. 58-520-071, 523-706, fax 58-522-168, 522-243. (ap)

Oprogramowanie dla administracji państwowej

Ministerstwo Pracy i Polityki Socjalnej podpisało 1 lipca br. kontrakt z polską firmą Computer Systems for Business International SA (CSBI) na opracowanie, wdrożenie i wykonanie do końca przyszłego roku, zintegrowanego, kompleksowego pakietu oprogramowania, przeznaczonego dla Krajowego Urzędu Pracy, Wojewódzkich Urzędów Pracy oraz 350 spośród ok. 500 Rejonowych Urzędów Pracy. Przedsięwzięcie to jest realizowane w ramach projektu automatyzacji organizacji zatrudnienia i pomocy społecznej ALSO (Automatization of the Labor and Social Welfare Organization), którego głównym kredytodawcą jest Bank Światowy. System ma umożliwić bardziej efektywne zarządzanie rynkiem pracy dzięki skróceniu czasu obiegu informacji, rozszerzeniu ich zakresu i szczegółowości oraz ułatwieniu dostępu do nich.

Zdaniem Dyrektora Biura ds. Informatyki MPiPS **Gustawa Pietrzyka** koszty oprogramowania, związane z realizacją projektu ALSO, stanowią 15% wartości ubiegłorocznej sprzedaży całego profesjonalnego software'u w Polsce. Główna część środków, pochodzących od kredytodawcy, będzie związana z realizacją kontraktu, podpisanego z CSBI SA.

Dostawca oprogramowania musiał spełniać bardzo rygorystyczne wymogi MPiPS oraz Banku Światowego, dotyczące potencjału osobowego, technologicznego i kapitałowego. Konieczne było wykazanie się zdolnością do przygotowania oprogramowania o standardzie światowym – skomentował wybór CSBI SA na wykonawcę oprogramowania Gustaw Pietrzyk.

Prezes CSBI SA **Roman Dolczewski** potwierdził gotowość do podjęcia tego poważnego zadania. Wyraził również zadowolenie ze współpracy ze specjalnym, stworzonym na potrzeby tego przedsięwzięcia, profesjonalnym zespołem resortu: *Udało nam się precyzyjnie opracować wszystkie założenia, ustalić harmonogram i warunki realizacji kontraktu.*

Realizacja projektu będzie przebiegać w trzech etapach:

1) przygotowanie szczegółowego projektu aplikacji,
2) przygotowanie i wdrożenie wersji pilotowej,
3) przygotowanie i wdrożenie wersji docelowej w pełnym zakresie.

Jest to największy kontrakt na dostawę oprogramowania dla administracji państwowej w tym roku. (ap)

Superkomputer za 14 mln dolarów

Laboratorium badań molekularnych w Waszyngtonie zamówiło najsilniejszy przetwarzający równolegle komputer (MPP), jaki IBM dotychczas wyprodukował. Komputer IBM RS/6000* Scalable POWERparallel Systems* (SP*), wartości 14 mln dolarów będzie przetwarzał dane i wykonywał obliczenia dla badań molekularnych. Został zamówiony przez Środowiskowe Laboratorium Badań Molekularnych (Environmental Molecular Sciences Laboratory – EMSL), które stanowi część słynnego Pacific Northwest National Laboratory (PNNL). PNNL jest jednym z dziewięciu laboratoriów departamentu energii USA. Wykonuje ono badania naukowe i technologiczne, dotyczące środowiska naturalnego o podstawowym znaczeniu dla kraju.

EMSL jest nowym laboratorium badawczym tego departamentu. Początkowo miało być ono przeznaczone do badań podstaw procesów molekularnych, niezbędnych do opracowania efektywnych i oszczędnych technologii unieszkodliwiania i długookresowego przechowywania niebezpiecznych odpadów promieniotwórczych w składowiskach podlegających departamentowi energii.

Symulacje komputerowe i modelowanie przeprowadzane na nowym komputerze pomogą zrozumieć na poziomie molekularnym przebieg procesów fizycznych, chemicznych i biologicznych, ważnych dla ochrony środowiska. Dzięki olbrzymiej mocy, potrzebnej do wykonywania złożonych obliczeń chemicznych, będzie wspomagał proces tworzenia nowych materiałów z odpadów i symulację biodegradacji odpadów organicznych, zastępując tym samym kosztowne i długotrwałe badania eksperymentalne.

Naukowcy z EMSL chcą stworzyć model wędrówki skażeń chemicznych i radioaktywnych spod powierzchni ziemi na zewnątrz skorupy ziemskiej i do powietrza. Obliczenia komputerowe będą symulować wędrówkę i reakcje tych zanieczyszczeń, co umożliwi optymalne projektowanie i przeprowadzanie akcji chroniących zagrożone środowisko naturalne.

Instalacja całego systemu komputerowego zostanie zakończona w marcu 1997 r. Komputer będzie miał wydajność ponad 200 GFLOPS (miliardów operacji w ciągu sekundy). Będzie się składał z 472 jednostek przetwarzających (węzłów), niezależnych, ale współpracujących ze sobą, 62 GB pamięci operacyjnej i 1,238 GB pamięci dyskowej.

Warto przypomnieć, że taki sam technologicznie komputer IBM RS/6000, ale o znacznie mniejszej liczbie jednostek zainstalowano we Wrocławiu. KBN zakupił mianowicie w ubiegłym roku komputer z 15 jednostkami przetwarzającymi, 1 GB pamięci operacyjnej i dyskami o pojemności 18 GB. (lw)

Meteorologiczny projekt EuroMET

European Meteorogical Education and Training Project – EuroMET (Europejski Projekt Meteorologicznej Edukacji i Szkoleń) ma ułatwić naukę meteorologi specjalistom i studentom wyższych uczelni. System nauczania opiera się na narzędziach World Wide Web z wykorzystaniem technik interaktywnych nauki i współczesnych środków edukacyjnych, jak wideo, dźwięku, animacji, obróbki obrazu i symulacji. Twórcy projektu przewidują wykorzystanie języka Java, dzięki czemu będzie można korzystać z systemu pracując na dowolnym komputerze lub stacji roboczej. Uniwersytet w Edynburgu i francuski narodowy serwis meteorologiczny, Meteo France, we współpracy z 20 innymi organizacjami, zamierzają stworzyć 240 modułów, z których będzie się składał ten system szkoleniowy. Ponadto każdy kraj może opracować i dołączyć do systemu szkolenia własną lokalną wersję modułu. (ap)

Projekt Sun'a dla rosyjskiej energetyki

Rosyjskie Ministerstwo Gospodarki Paliwowo-Energetycznej wybrało Sun Microsystems Computer Company na wykonawcę terabajtowego systemu rozproszonej bazy danych z centrami w Moskwie i regionach przemysłu paliwowego we Wspólnocie Niepodległych Państw. Projekt, którego realizacja ma potrwać pięć lat, będzie kosztował od 50 do 200 mln dolarów.

System ma tworzyć jednolitą infrastrukturę informacyjną, zapewniającą przedsiębiorstwom naftowym oraz agendom rządowym WNP dostęp do danych o wydobyciu i produkcji ropy naftowej. Dane będą pobierane z ok. 1000 pól naftowych znajdujących się na terenie WNP.

W tym projekcie będą wykorzystane dane o zasobach naturalnych i komputerowe standardy, udostępnione przez przemysłowe organizacje typu non-profit, jak np. Petrotechnical Open Software Corporation.

Bazą techniczną projektu będą serwery Suna Ultra Enterprise, SPARCserver i stacje robocze SPARCstation. System zarządzania bazami danych dostarczy Sybase.

Przy okazji warto dodać, że w Moskwie pracuje zespół Borysa Babayana, który uczestniczy w programach badawczych Suna m.in. w zakresie zaawansowanych metod kompilacji, a prywatna firma ELVIS+ przygotowuje wspólnie z amerykańskim partnerem nową technologię dla sieci bezprzewodowych. (lw)

Brytyjski przewodnik ochrony komputerowych informacji

Według Ian'a Taylor'a ministra Nauki i Techniki Wielkiej Brytanii **koszty awarii w systemach informacji komputerowej wynoszą przeciętnie w firmie tego kraju ok. 16.000 funtów rocznie.** Niektóre firmy bagatelizują problem bezpieczeństwa informacji przechowywanych w ich komputerach. Nie bardzo zdają sobie sprawę, że trzeba nimi specjalnie zarządzać, chronić je przed kradzieżą czy uszkodzeniem, tak jak każdą inną wartościową rzecz należącą do firmy.

Problem ten dostrzegł Department of Trade & Industry (Departament Handlu i Przemysłu) brytyjskiego rządu i sfinansował wydanie podręcznika omawiającego zagadnienia zabezpieczania danych komputerowych. Popularyzacja tego podręcznika ma wpłynąć na zmniejszenie strat w gospodarce, spowodowanych niewiedzą lub niefrasobliwością zarządów firm brytyjskich. (ap)

Wspólna oferta Informix, HP i Netscape

Informix, Hewlett-Packard i Netscape uzgodniły, że Informix będzie pierwszym producentem relacyjnych baz danych, dostosowanych do przyszłych produktów HP i Netscape, obsługujących pracę grupową w otwartej strukturze sieci typu intranet.

Wewnętrzne sieci intranet są wzorowane na standardach Internetu, ale z punktu widzenia nakładów inwestycyjnych i kosztów utrzymania są dużo efektywniejszym sposobem rozprowadzania informacji w geograficznie rozproszonych przedsiębiorstwach.

Informix, dzięki udostępnieniu technologii dynamicznie skalowalnych baz danych, umożliwi użytkownikom sieci intranet łatwy dostęp do informacji. Technologia ta pozwoli również na prowadzenie zespołowych prac nad dokumentami i projektami. Rozwiązania Informixa mogą być stosowane zarówno w aplikacjach typu klient/serwer, jak i w sieciach intranet. Dzięki temu możliwe jest rozwiązanie integrujące informację przetwarzaną i przechowywaną w relacyjnych bazach danych, np. aplikacje OI.TP i hurtownie danych.

Informix ma dostosować się do przemysłowego standardu HP OpenView, stosowanego do zarządzania rozproszonymi sieciami, systemami, aplikacjami i bazami danych. (ap)

Obraźliwe synonimy

W odpowiedzi na publikacje prasowe, dotyczące synonimów o rasistowskiej wymowie, które pojawiły się w hiszpańskojęzycznym słowniku wyrazów bliskoznacznych (Tezaurus), Microsoft postanowił zweryfikować polską wersję słownika, aby uniknąć podobnych kłopotów.

Dostawcą polskiej wersji słownika jest firma TIP Sp. z o.o.. Został on oparty na „Słowniku Synonimów" Andrzeja Dąbrówki, Ewy Geller i Ryszarda Turczyna, wydanym w 1993 r. W tym słowniku także znaleziono zestawy synonimów, które mogą być odebrane jako obraźliwe dla niektórych grup etnicznych lub narodowości.

Nowa wersja słownika, natychmiast po jego weryfikacji i opracowaniu, zostanie udostępniona wszystkim użytkownikom dotychczasowej wersji Tezaurusa na stronie internetowej Microsoft: http://www.Microsoft.com/Poland/. (ap)

Kodowanie w Internecie

Microsoft w środowisku Windows i aplikacjach w nim pracujących stopniowo przechodzi ze stron kodowych ANSI na ogólnoświatowy standard, określony normą ISO 10646-1, którego część dotycząca pierwszych znormalizowanych 64 tys. znaków znana jest pod nazwą Unicode.

Produkty Microsoft działające w środowisku internetu są systematycznie dostosowywane do współpracy ze stronami kodowymi ISO, oznaczonymi numerami 8895-1, 8895-2, itd.

Programy takie jak: Microsoft Explorer 2.0 CI PL., Microsoft Internet Explorer 3.0, Mail and Newsreader, Microsoft Exchange oraz systemy operacyjne Windows mają wbudowane kodowanie w standardzie Unicode. Przyszłe produkty Microsoft będą także do niego dostosowywane. (ap)

Telefon komórkowy z interfejsem modemowym i faksowym

Telefon przenośny Alcatel HC1000 jako pierwszy model serii HC firmy Alcatel Telecom został wyposażony w interfejs modemowy i faksowy. Współpracuje on z komputerem typu laptop, za pomocą którego można odbierać i przesyłać dane i faksy bez użycia papieru z prędkością do 9600 bps. Telefon ma wymiary 143 × 60 × 23 mm, objętość 170 cm i waży 180 g. Podłączenie do sieci jest możliwe dzięki karcie SIM wielkości karty kredytowej. Można na niej zapisać 100 numerów telefonicznych. Baterie NiMH pozwalają na rozmowę przez 140 min. lub na 42 godziny pracy w stanie stand-by. Alcatel HC 1000 posiada wyświetlacz ciekłokrystaliczny o dużej powierzchni; pięć wierszy po 15 znaków. (kar)

Systemy czasu rzeczywistego

III Konferencja *Systemy Czasu Rzeczywistego*, zorganizowana przez Wydziałowy Zakład Informatyki Politechniki Wrocławskiej oraz Instytut Automatyki Systemów Energetycznych we Wrocławiu odbyła się w dniach 10-13.09 w Szklarskiej Porębie. Na czele Komitetu Organizacyjnego stał prof. Zbigniew Huzar.

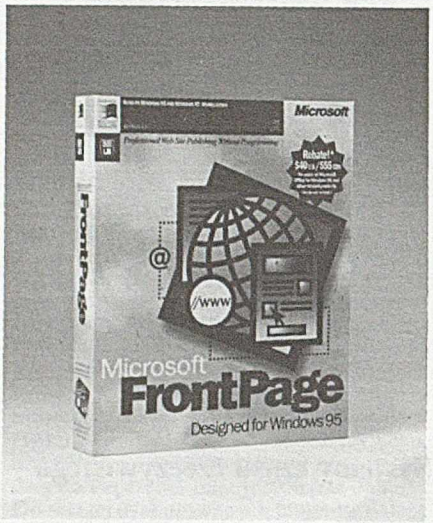
Uczestnicy konferencji głównie zajmowali się problemem **bezpieczeństwa oprogramowania**.

Zaprezentowano referaty dotyczące m. in.: analizy i modelowania systemów czasu rzeczywistego, niezawodności oprogramowania, specyfikacji, projektowania i implementacji oprogramowania oraz zastosowań różnych systemów w przemyśle. (ap)

The Fifth International Conference On Information Systems Development

24-26.09 odbędzie się w Sopocie międzynarodowa konferencja na temat budowy systemów informatycznych, organizowana przez Wydział Systemów Informatycznych Uniwersytetu Gdańskiego. W konferencji uczestniczyć będą specjaliści z całego świata. Tematyka obejmuje szeroki zakres zagadnień od podstaw teoretycznych budowy systemów informacyjnych poprzez systemy baz danych, inżynierię procesów biznesowych (BPR), pracę grupową do inżynierii programowania. (lw)

Tworzenie stron WWW – lekko, łatwo i przyjemnie



W drugiej połowie sierpnia w Warszawie Microsoft zaprezentował program FrontPage 1.1, który sprawia, że tworzenie stron WWW w Internecie przestaje być domeną programistów i specjalistów, znających składnię języka hipertekstowego HTML. Każdy użytkownik Internetu wykorzystując FrontPage 1.1 może z łatwością stworzyć własne strony WWW w Internecie lub Intranecie i odpowiednio nimi zarządzać. Microsoft FrontPage Editor pozwala na tworzenie i edycje stron WWW w trybie WYSIWYG (to, co widzisz na ekranie, otrzymujesz w dokumencie końcowym), bowiem program automatycznie generuje poprawne kody HTML. Ponadto importuje i konwertuje istniejące pliki generujące HTML 2.0 oraz najnowsze style HTML 3.0. Specjalny edytor pozwala również na tworzenie graficznej mapy z ukrytymi przejściami. W podobny sposób odbywa się tworzenie odnośników do innych stron WWW – wystarczy przeciągnąć i upuścić lub kliknąć przycisk, by utworzyć połączenie. Do programu wbudowano też narzędzia do tworzenia formularzy z polami tekstowymi, polami wyboru, przyciskami radiowymi oraz rozwijanymi listami. W FrontPage 1.1 znajdują się Kreatorzy oraz szablony pozwalające na tworzenie pojedynczych stron lub całych węzłów. Zautomatyzowano wiele funkcji przez komponenty WebBot realizujące pewne funkcje, które normalnie wymagają pisania skryptów CGI. Komponenty WebBot są umieszczone na stronie i pozwalają na dodanie funkcji interaktywnych, jak listy dyskusyjne, tekstowe mechanizmy wyszukiwania oraz obsługę formularzy. FrontPage 1.1 należy do pakietu Microsoft Office i jest w pełni zintegrowany z pozostałymi aplikacjami grupy. Narzędzia programu pozwalają na skuteczną kontrolę i zarządzanie serwerem. Serwer WWW może być umieszczany na komputerach działających w systemie Windows NT, Windows 95 i kilku odmianach systemu Unix.

Modem światłowodowy E1/T1

Firma RAD Data Communications wprowadza na rynek nowy modem światłowodowy FOM-E1/T1, który zapewnia tanią transmisję wysokiej jakości w paśmie E1 lub T1, przez łącza światłowodowe. Modem przekształca elektryczne sygnały E1 lub T1 na sygnały światłowodowe i odwrotnie, dzięki czemu istniejące urządzenia E1 lub T1 mogą wykorzystywać zalety łączy światłowodowych. Z opcjonalną diodą laserową modem może przesyłać dane na odległość do 50 km. Wyposażony jest w zrównoważone i niezrównoważone interfejsy do E1 i ma optyczne interfejsy ST, SMA lub FC do jednomodemowego i wielomodemowego kabla światłowodowego. Urządzenie zapewnia też dwupoziomową transmisję optyczną, dzięki czemu może współpracować z optycznymi wtórnikami stosowanymi na liniach światłowodowych. Oprócz modelu wolnostojącego RAD dostarcza modemy FOM-E1/T1 w postaci karty montowanej w stojaku modemowym ASM-MN-214. Wersja modułowa nadaje się do lokalnych zastosowań rozproszonych lub centralnego węzła sieci korporacyjnej; stojak może pomieścić do 14 kart.

GroupWise WebAccess dla sieci Intranet

Dostępne jest już oprogramowanie Novella dla sieci Intranet – GroupWise Welb Access, które oferuje użytkownikom systemu dostęp do Universal In Box (uniwersalnej skrzynki wejściowej) z dowolnego miejsca i o każdej porze za pośrednictwem Internetu i przy użyciu dowolnej przeglądarki sieciowej zgodnej z HTML 1.0. GroupWise WebAccess jest jedynym produktem na rynku, który umożliwia przesyłanie i otrzymywanie przez Internet poczty elektronicznej i głosowej, faksów, informacji o terminach spotkań i zadaniach do wykonania za pośrednictwem jednego interfejsu. Produkt jest dostępny dla użytkowników systemu GroupWise 4.1, a do końca br. będzie dostępny również dla użytkowników systemu GroupWise 5. Oprogramowanie GroupWise Web Access umożliwia użytkownikom systemu GroupWise, którzy znaczną cześć pracy wykonują korzystając z Internetu, połączenie z wewnętrzną siecią komputerową firmy, uczestniczenie w różnych zadaniach, kontrolowanie stanu ich zaawansowania oraz zarządzanie własnymi terminarzami bez konieczności przerywania pracy z przeglądarką sieciową. Oprogramowanie jest sprzedawane w 18 różnych wersjach językowych i obsługuje najpopularniejsze systemy operacyjne i sprzętowe.

Kompleksowy system do zarządzania biurami maklerskimi

Największy polski producent oprogramowania – gdyńska firma Prokom – wdrożyła w Domu Maklerskim Banku Zachodniego nową wersję własnego, kompleksowego systemu do zarządzania biurami maklerskimi Promak 2.0. Obecnie wdraża go w biurze Maklerskim Banku Gdańskiego. System umożliwił połączenie wszystkich 36 Punktów Obsługi Klienta z centralą za pomocą sieci rozległej Telbank i awaryjnie, przez łącze satelitarne V-SAT. W skład systemu wchodzą dwa serwery unixowe: większy bazodanowy (HP T500-4CPU) z Informix Dynamic Server 7.0 oraz mniejszy aplikacyjny (HP k200-2CPU) wraz z interfejsem użytkownika oraz 100 końcówek komputerów PC z zainstalowanym emulatorem terminala. Dostęp do centralnej bazy danych jest możliwy w trybie on-line za pomocą interfejsu znakowego z każdego Punktu. W przypadku awarii jednego z serwerów drugi może przejąć jego funkcje. Nowa wersja systemu zapewnia kompleksową obsługę biura maklerskiego. Prokom zapowiada, że jeszcze w tym roku ukaże się kolejna wersja systemu.

ATM 25 Mb/s z FORE Systems

Firma FORE Systems, znana z produkcji urządzeń sieciowych standardu ATM (Asynchronous Transfer Mode), wprowadziła na rynek (w Polsce za pośrednictwem dystrybutora – firmy ATM) tanie i kompletne rozwiązanie dla sieci standardu ATM o przepustowości 25 Mb/s. ATM ForeRunner 25 Mb/s obejmuje przełącznik ATM ForeRunner, moduły interfejsów, karty sieciowe ATM ForeRunnerLE oraz oprogramowanie ForeThought 4.0, zgodne ze standardem LAN Emulation 1.0, zapewniające współpracę z wirtualnymi sieciami lokalnymi. Urządzenia ATM ForeRunner 25 Mb/s mogą współpracować z okablowaniem galwanicznym UTP kategorii 3,4 lub 5, co pozwala klientom wykorzystać już istniejące instalacje. Zgodne są też z całą linią produktów FORE i współpracują z oprogramowaniem do zarządzania siecią – ForeView.

Merlin z Javą

IBM zapowiedział, że do następnej wersji systemu operacyjnego OS/2 Warp, zwanej roboczo Merlin, przyłączona będzie nie tylko przeglądarka WWW, osadzona w środowisku Java firmy Sun Microsystem, ale także środowisko programowania Java. Merlin pojawi się więc na rynku z wbudowanym zestawem Java Developers Kit oraz modułem run-time, niezbędnym do uruchamiania aplikacji pisanych w języku Java. Oznacza to, że użytkownicy systemu OS/2 będą mogli ściągać z Internetu programy pisane w języku Java, a następnie umieszczać je w swoich komputerach, a także osadzać w programach, np. arkuszach kalkulacyjnych lub edytorach tekstu. Dzięki interfejsowi użytkownika, ukierunkowanemu na Internet oraz wbudowanemu oprogramowaniu do rozpoznawania mowy, użytkownicy Merlina będą mogli przeciągać programy pisane w języku Java ze strony WWW na pulpit, a następnie uruchamiać je podwójnym kliknięciem myszy lub poleceniem głosowym. Rozpoznawanie mowy będzie możliwe w następujących językach: angielskim (z wymową amerykańską i brytyjską), niemieckim, francuskim, włoskim i hiszpańskim.

Firmy: Apple Computer, IBM, Netscape, Oracle oraz Sun ogłosiły wspólny zbiór wytycznych do rozwijania tanich, łatwych w użyciu urządzeń do przetwarzania sieciowego. Celem stworzenia zbioru wymagań technicznych, znanego pod nazwą „NC Reference Profile 1" jest upowszechnienie multimedialnego przetwarzania w Internecie. Dokument nie narzuca architektury, ale ma na celu ułatwienie rozwoju branży przetwarzania sieciowego, z jednoczesną ochroną inwestycji poczynionych przez klientów, producentów wyposażenia, dostawców systemów oraz usług serwisowych i aplikacji przez zgodność produktów, która obejmuje całą branżę.

*

Novell Polska odnotował znaczny wzrost zainteresowania umowami licencyjnymi Volume License Agreement (VLA). Tylko w lipcu skorzystało z tej oferty 30 firm. Główne korzyści, wynikające z zawarcia umowy VLA, to możliwość kupowania produktów Novella po niższych cenach oraz swobodniejsze dysponowanie licencjami na zakupione produkty. Upust cenowy może wynieść nawet 37%.

*

Na mocy umowy generalnej, zawartej na dwa lata pomiędzy Computerland Poland SA i Telewizją Polską SA, Telewizja będzie kupować sprzęt komputerowy firmy COMPAQ za pośrednictwem Computerlandu. Umowa przewiduje m.in. zakup najnowszych wieloprocesorowych serwerów ProLiant 5000 z procesorem Pentium Pro, kilkuset stacji roboczych oraz notebooków. Compaq podejmuje z Oraclem współpracę, mającą na celu optymalizację systemu Oracle Universal Server dla Compaq ProLiant 5000.

*

Według raportu IDC firma ACER w I kwartale br. osiągnęła czwartą pozycję na świecie pod względem ilości sprzedanych serwerów (254 tys. sztuk); w 1995 r. Acer zajmował piątą pozycję. Najnowszą ofertą firmy jest Aceraltos 19000 z dwoma procesorami Pentium Pro o częstotliwości 66 MHz. Szyna PPro-MPBus umożliwia transmisję danych z szybkością 533 MB/s.

*

Firma Oracle informuje, iż dostępne jest już oprogramowanie Oracle InterOffice, w pełni skalowalne, oparte na technologii WWW do pracy grupowej. Może ono być skalowalne w zależności od pracy w grupach roboczych aż do pracy w ramach całego przedsiębiorstwa. Oracle Inter-Office pozwala na współpracę tysięcy jednoczesnych użytkowników z jednym serwerem.

*

American Power Conversion (APC) wprowadziła do sprzedaży nowy moduł rozszerzający z rodziny SmartSlot, pod nazwą SmartSlot Relay I/O Module. Umożliwia on użytkownikom kontrolowanie oraz monitorowanie zasilania z wykorzystaniem interfejsu w postaci standardowego 25-pinowego gniazda połączeniowego. Gniazdo to jest używane np. jako port szeregowy w technice komputerowej, połączeniach telekomunikacyjnych, centralach abonenckich oraz systemach alarmowych.

*

IBM rozpoczął sprzedaż pakietu VoiceType 3.0 dla Windows 95, umożliwiającego użytkownikom komunikację z komputerem za pomocą głosu. Jest on przydatny szczególnie dla użytkowników przetwarzających duże ilości tekstu oraz dla osób, których zadania wymagają obsługi komputera bez używania dłoni. VoiceType IBM cechuje połączenie ciągłości sterowania z możliwością dyktowania z przerwami, przy jednoczesnym zapewnieniu niezależności od osoby mówiącej. Teksty można dyktować bezpośrednio do najczęściej używanych aplikacji, takich jak Word, WordPerfect czy Lotus Notes.

*

Wszyscy, którzy kupią nowe stacje robocze firmy Dell – Dell Optiplex (z procesorem Pentium Pro) z zainstalowanym systemem operacyjnym Windows NT Workstation 3.51 mają zapewniony bezpłatny upgrade do wersji 4.0.

*

Intel i Microsoft podpisały szerokie porozumienie licencyjne, którego celem jest umożliwienie milionom ludzi łatwego przesyłania głosu, wideo i danych w trakcie połączeń konferencyjnych. Intel udostępni Microsoftowi swoją implementację standardów H.323, RSVP i RTP, opracowanych do serii wyrobów wideokonferencyjnych ProShare, natomiast Microsoft udostępni Intelowi swoją implementację T.120, Technologię ActiveX oraz aplikację Microsoft NetMeeting. Obie firmy współpracują z producentami internetowych książek telefonicznych i usługodawcami, nad ułatwieniem wyszukiwania za pomocą User Location Service (ULS) osób, z którymi chcemy porozmawiać przez Internet. Microsoft udostępnia Intelowi swoją implementację ULS.

*

Znany polski dystrybutor – firma California Computer – wprowadziła do swojej oferty produkty sprzedane pod własną nazwą. W pierwszym etapie są to płyty główne i karty graficzne, które będą produkowane według specyfikacji opracowanej przez California Computer. Firma brała udział także w procesie opracowywania i testowania produktów.

*

Warszawska firma SIMPLE wygrała przetarg na system zarządzania Biblioteką Publiczną m.st. Warszawy. SIMPLE dostarczy w ramach kontraktu oprogramowanie do zarządzania kadrami, prowadzenia finansów i księgowości oraz gospodarki magazynowej. Wdrożenie będzie przebiegało w trzech etapach. W pierwszym i drugim, trwającym po trzy miesiące, SIMPLE wdroży moduł do zarządzania kadrami oraz najnowszą wersję modułu finansowo-księgowego w wersji 4.5. W ostatnim etapie zostanie wdrożony moduł magazynowy. Cały system ma być wdrożony przed końcem br.

*

W ramach działającego na Politechnice Gdańskiej Centrum Edukacyjnego Uczelnianej Sieci Komputerowej uruchomiono sekcję zajmującą się przygotowaniem sieciowego szkolenia nt. Obsługa Programu AutoCAD. Szkolenie to, jeszcze w fazie prototypu, jest przeznaczone przede wszystkim dla pracowników i studentów uczelni, którzy chcą skorzystać z eksternistycznej formuły kursu. Informacja o szkoleniu jest dostępna pod adresem: WWW http://www.pg.gda.pl/ dec. Rejestracja uczestników odbywa się via e-mail na adres: blanka@pg.gda.plub dec@pg.gda.pl.

*

Firma Sequiter Software Inc. opracowała kolejną wersję biblioteki do zarządzania bazami danych xBASE dla języków C, C++, VisualBasic i Delphi – CodeBase 6. Biblioteki do wszystkich tych języków wraz z kodem źródłowym i plikami pomocy znajdują się na jednej płycie CD-ROM. CodeBase obsługuje dane w formatach dBASE, Clipper, FoxPro, Visual dBASE i Visual FoxPro wraz z indeksami i plikami memo, możliwe jest więc równoczesne korzystanie z tych samych danych przez program np. FoxPro, jak i jego odpowiednik w C++, korzystający z procedur pakietu CodeBase.

*

Sun Microsystems wprowadził do sprzedaży serwer Netra NFS 150, w którym zastosowano nową architekturę serwerów Sun Ultra. Twórcy serwera położyli nacisk na optymalne przystosowanie go do potrzeb użytkowników korzystających ze zbiorów za pośrednictwem sieci, a więc przede wszystkim do szybkości. Zainstalowane wcześniej oprogramowanie pozwala na przygotowanie Netra NFS do pracy w ciągu zaledwie 30 minut, zaś interfejs użytkowy oparty o standard HTML zapewnia bardzo prostą obsługę.

*

Firma 3Com wprowadziła do sprzedaży LinkBuilder FMS 100 – pierwsze urządzenie zapewniające obsługę sieci zgodną z architekturą SNMP (Simple Network Management Protocol), przeznaczone dla grup roboczych 100BASE-T. Jednostka Lin Builder FMS 100 jest podłączona do zestawu siedmiu koncentratorów, zapewniając obsługę zestawu w pełni zgodną z architekturą SNMP.

Oprac.:
Krystyna Karwicka-Rychlewicz

Zaprosili nas...

Computer Associate na konferencję CA World 96' w Nowym Orleanie

ComputerLand Poland S.A. oraz Digital Equipment Polska Sp. z o.o. na konferencję prasową na temat współpracy między obiema firmami

IBM na konferencję „IBM OS/390 i S/390: Informatyka XXI wieku" połączoną z prezentacją systemu

Microsoft na konferencję prasową i pokaz programu do tworzenia i zarządzania stronami WWW **Microsoft FrontPage 1.1** oraz premierą przeglądarki internetowej **Microsoft Internet Explorer 3.0**

Aplikacja zdrowego rozsądku

Z **Markiem Gondzio**, prezesem Stowarzyszenia Polski Rynek Oprogramowania rozmawiają Lesław Wawrzonek i Agnieszka Papis.

– Jakie zamierzenia PRO wiążą się z powołaniem nowego Zarządu?

– Nie chcę tworzyć wrażenia, że budując nową strategię i nowy wizerunek PRO odcinamy się od tego, co było wcześniej – będzie to pewnego rodzaju kontynuacja. Przede wszystkim chcemy bardziej integrować szeroko rozumiane środowisko informatyczne. Będziemy też systematyzować i zdecydowanie wzmacniać działalność promocyjno-informacyjną. Zmierzamy do tego, aby stowarzyszenie było postrzegane nie tylko jako strona i biegły w sprawach sądowych, czyli w momencie, gdy prawa autorskie już zostały naruszone, ale będziemy też współpracować przy tworzeniu tego prawa. Na przykład zajęliśmy stanowisko w sprawie ustawy o samorządzie i ustawy o zamówieniach publicznych. Na pewno będziemy uczestniczyć w tworzeniu ustawy o ochronie danych osobowych i dokumentu elektronicznego.

Gromadzimy też informacje prasowe i korespondencję, dotyczące sfery sprzedaży oprogramowania. Chcemy dostarczać jak najwięcej informacji naszym członkom i osobom zainteresowanym danymi o oprogramowaniu. Zależy nam przede wszystkim na tym, żebyśmy się znali i szanowali nawzajem, nawet jeżeli pracujemy w konkurencyjnych firmach i dostrzegali, że mamy wiele wspólnych interesów.

– Niedawno odbyła się konferencja Microsoftu i Business Software Aliance SA na temat piractwa komputerowego Polsce. Czy Pan słyszał coś o tym?

– Słyszałem.

– Ale nie został pan oficjalnie zaproszony?

– Akurat o tym spotkaniu nie byłem powiadomiony. Sądzę jednak, że stopniowo nasze relacje z BSA będą się normalizować. Rozmawiałem niedawno z dyr. Sielskim z Microsoftu – wiodącego przedstawiciela BSA – i uzgodniliśmy zasady współpracy, a nie rywalizacji. Na przykład na targach CAD/CAM w tym roku prawnik stowarzyszenia PRO dyżurował i udzielał porad, gościnnie w stoisku Autodesku, który też należy do BSA. Chcielibyśmy też, żeby ktoś z Microsoftu bezpośrednio poinformował naszych członków o tych systemach sprzedaży ich produktów, które nie są typowe dla polskiego rynku, np. umowy typu Select. Ponadto spotykamy się na tych samych imprezach, np. na szkoleniu policji i prokuratorów we Wrocławiu, organizowanym przez Komendę Główną Policji. Nasze działania się uzupełniają i nie dostrzegam w tej chwili potencjalnych źródeł konfliktu.

dr Marek Gondzio - 40 lat, ukończył Wydział Elektroniki Politechniki Warszawskiej. Pracował w Instytucie Informatyki na Politechnice Warszawskiej 11 lat, a od 6 lat jest wicedyrektorem firmy InterDesign, zajmującej się m.in. produkcją oprogramowania inżynierskiego do obróbki wielkoformatowych dokumentów skanowanych. Od kwietnia 1996 r. jest prezesem zarządu Stowarzyszenia PRO

– Czy ktoś, kto orientuje się w problemach piractwa komputerowego, nie odnosi wrażenia, że społeczeństwo otrzymuje wypaczony obraz tego problemu, gdy korzysta z informacji podawanych przez popularne media?

– Niedawno telewizyjna Panorama informowała o tym, że nikt nie chroni twórców oprogramowania. Takie informacje można oczywiście weryfikować, ale ja nie mam złudzeń co do tego, że sprostowanie zostanie opublikowane. PRO dostarcza regularne informacje dla prasy. Jesteśmy obecni na targach, np. przedstawimy swoje referaty na Softargu'96. Niestety nie wszystkie nasze doniesienia trafiają na łamy gazet, mimo że niektóre są bardzo interesujące, choć nie szokujące. Mój zarzut dotyczy nie tylko prasy popularnej, ale także fachowej, która chyba nazbyt goni za sensacjami. Mam nadzieję, że niedługo ta sytuacja się zmieni, licząc także na „Informatykę".

– A co z pomysłem napisania raportu o oprogramowaniu w Polsce? W jaki sposób chciałby Pan to zrobić?

– Niestety przeciąga nam się to w czasie. Nagromadziliśmy sporo materiałów, zestawień tematycznych, branżowych i rankingów firm. Warto byłoby też wiedzieć, co o ochronie oprogramowania sądzą jego producenci i twórcy: czy traktują to jako rzecz cenną, ważną, czy uwzględniają to w swojej działalności czy też na razie godzą się z piractwem. Jak wygląda nasz rynek aplikacji i integracji? Jakie przepisy prawne sprzyjają, a jakie przeszkadzają rozwojowi rynku oprogramowania w Polsce? Ile wartości dodają do programów z półki polscy informatycy i na czym się koncentrują? Jest wiele takich pytań, na które warto byłoby znać odpowiedź. Myślę, że byłaby ona ciekawa nie tylko dla naszego środowiska, ale również dla sfer rządowych, w tym dla Komitetu Badań Naukowych. Być może po przyjrzeniu się polskiemu rykowi oprogramowania nastąpi weryfikacja listy priorytetowych dziedzin i zjawisk, wymagających nowych regulacji prawnych.

– Ale jest to przedsięwzięcie bardzo kosztowne...

– Na pewno, a PRO nie byłoby w stanie go sfinansować. W tej chwili otrzymuję sygnały, że jest możliwość zbudowania nieco szerszej koalicji dla realizacji tego celu. Zainteresowaliśmy nim środowisko informatyczne, prasę, Polską Izbę Informatyki i Telekomunikacji oraz środowiska uczelniane. Niestety jest to wciąż jeszcze nie zrealizowany pomysł, którego ramy organizacyjne właśnie opracowujemy.

– W jaki sposób PRO współpracuje z tymi firmami, które zarejestrowały w Stowarzyszeniu swoje produkty? Czy to polega wyłącznie na angażowaniu się w momencie konfliktu?

– Możliwości naszej działalności są rozmaite. Znajo-

mość prawa autorskiego w ogólności, a w odniesieniu do oprogramowania w szczególności jest jeszcze bardzo słaba w środowisku policji i wymiaru sprawiedliwości. Procesy sądowe trwają zbyt długo. Następują oczywiste naruszenia prawa autorskiego, a pozwani, żeby uciec od odpowiedzialności, stosują mnóstwo kruczków prawnych. Jeżeli do sądu czy prokuratury trafia sprawa o oprogramowanie, które zostało zarejestrowane w PRO, to sprawa przedstawia się w całkowicie innym wymiarze niż w przypadku, gdy właściciel programu mówi prokuratorowi: „Wydaje mi się, że ukradli mi program". Program, który jest zarejestrowany, a zarządzaniem jego prawami autorskimi zajmuje się organizacja, która ma do tego tytuł, wynikający z zezwolenia ministra kultury i sztuki, jest traktowany zupełnie inaczej i w prokuraturze i w sądzie. Naszą rolą jest także nieustanne dostarczanie informacji policji i prokuraturze. Proszę sobie wyobrazić przesłuchanie podejrzanego o kradzież oprogramowania – po dwóch stronach stołu siedzą ludzie o diametralnie różnej wiedzy na temat tego zagadnienia. Policjant, który często nie wie w ogóle, co to jest program i podejrzany, który doskonale się na tym zna. Niedawno na spotkaniu z sejmową Komisją Kultury jeden z zebranych przytaczał liczbę spraw o naruszenie praw autorskich. Sądziłem, że będzie ich tysiące, a okazało się że liczba ta nie przekracza kilkuset, a dwadzieścia parę spośród nich dotyczy oprogramowania. Dodajmy, że jeżeli sprawa trwa trzy lata, to dla pokrzywdzonego sprawa traci aktualność, bo tej wersji programu najczęściej już nie ma na rynku.

– Przyglądając się ofertom sprzedaży oprogramowania trzeba być naprawdę dobrym ekspertem, żeby się zorientować, ile właściwie kosztuje dany program. Są tam różnego rodzaju upusty, sprzedaż promocyjna i wiązana, inne ceny dla nabywców programów konkurencji, upgrade'y o skomplikowanych algorytmach. Co jest powodem takiego zamętu w cennikach? Czy jest to z korzyścią dla klienta czy sprzedawcy?

– Czy podtekst pańskiego pytania był taki, że na tym rynku zamieszanie wywołano celowo, aby zdezorientować klientów i użytkowników? Wręcz przeciwnie – wszyscy próbują to uporządkować, tylko nie dają sobie z tym rady. Na pewno ten zamęt nie jest wywoływany specjalnie. Byłoby to zupełnie nieracjonalne, ponieważ wszystkim firmom zależy na tym, żeby systemy sprzedaży służyły osiąganiu dużych obrotów i tym samym – dużych zysków. Wynika on raczej z nietrywialnej dynamiki zjawisk, jakie zachodzą na rynku software'owym, w którym nowe produkty powstają codziennie. Wszystko tu dzieje się stosunkowo szybko i w warunkach ostrej konkurencji. Rzeczywiście oferta jest szeroka, istnieje dużo wariantów sposobu zakupu produktów i dlatego niektórym klientom trudno jest się zorientować. Co gorsza sądzę, że i firmom dilerskim sporo trudności przysparza nadążanie za zmianami w cenniku, zwłaszcza jeżeli mają w ofercie szeroką gamę produktów. Są sytuacje, że sprzedawca wie o produktach naprawdę niewiele.

– Dynamika rynku oprogramowania odbija się jednak negatywnie na klientach. Jeżeli nie można nigdzie w Polsce kupić programu pracującego na starszych typach komputerów, to czy dając komuś kopię takiego programu popełnia się przestępstwo?

– Tak, to nie podlega żadnej dyskusji. Niestety można kupić tylko to, co jest na rynku dostępne. Zapotrzebowanie na starsze wersje dowodzi tylko tego, że firmy dilerskie w Polsce nie dostrzegają tej niszy rynkowej. Jestem przekonany, że np. WordPerfecta PL 5.1 w dalszym ciągu można zamówić, gdyby się o to tylko jakiś diler postarał.

– Niektórzy producenci oprogramowania zmieniając swoją strategię rozwoju nie zajmują się tymi użytkownikami, którzy wcześniej nabyli ich produkty i liczą na dalszą współpracę w ramach umowy licencyjnej.

– Można tu mówić o dwóch aspektach. Pierwszy polega na tym, że producenci oprogramowania po podjęciu decyzji o zaprzestaniu produkcji jakiegoś artykułu na ogół mają do zaproponowania jakąś alternatywę. Być może nie wszyscy użytkownicy z niej korzystają i może nie każdy ją dostrzega.

Drugim aspektem jest rywalizacja – w tej branży kilka tygodni opóźnienia to być albo nie być. Nie wszystkie firmy są Microsoftem lub Novellem. Powstaje tysiące innych firm, które jak gwiazdy rozbłyskują na tym firmamencie i stają się potentatami, ale jest tysiące innych małych gwiazdeczek, które zabłysły i zgasły. Ci, którzy kupili produkty tych firm, mają prawdziwe problemy. Niewiele można na to poradzić.

– A więc kupując oprogramowanie należy brać pod uwagę pozycję finansową i możliwość przetrwania danej firmy, a nie tylko jakość programu i jego możliwości.

– To dotyczy niemal wszystkich dziedzin przemysłu. W mniejszym stopniu producentów żywności, bo po skonsumowaniu np. sera można nie przejmować się tym, czy będzie nadal produkowany i czy będzie miał następną wersję. Gdy produkty mają nieco trwalszy charakter, tak jak np. oprogramowanie, sprawa wygląda inaczej. Potrzeba trochę wyrozumiałości dla tego, co się w tej dziedzinie dzieje.

– Wiele osób jednak zauważa dokuczliwy brak lobby klientów, naciskającego na producentów technologii informatycznych.

– Takiego lobby istotnie nie ma, choć zaczynają się ostatnio w Polsce tworzyć. Mógłbym wskazać dwa takie środowiska, które zajmują się sprawami klientów. Jest to Stowarzyszenie Użytkowników Systemów Inżynierskich, do którego należy m.in. moja firma i wielu innych producentów. Jest też Polska Społeczność Internetu.

Klienci przyzwyczaili się już do tego, że błędy są nieodłączną częścią oprogramowania. Faktem jest, że często błędów jest zbyt dużo i oprogramowanie lub dokumentacja są wykonywane po prostu niechlujnie.

– Wiadomo, że koszt oprogramowania do obsługi dużej firmy jest znacznie niższy niż koszt jego wdrożenia. Czy potwierdza Pan przesunięcie zysków z firm produkujących oprogramowanie do firm zajmujących się wdrażaniem i integracją systemów?

– Opudełkowane oprogramowanie tanieje, spada także jego udział w globalnych obrotach rynku software'owego. Działalność integracyjno-aplikacyjna odgrywa coraz większą rolę, ponieważ potrzeby klientów są bardzo często specyficzne. Tu pojawia się wielka szansa dla naszego rodzimego środowiska informatycznego w takich dziedzinach jak: zastosowanie technologii MRP, systemy inżynierskie, wdrażanie oprogramowania ekonomicznego, dostosowywanie standardowych rozwiązań informatycznych do polskiego prawa i zwyczajów czy integracja narzędzi różnych producentów. Jest tu dużo miejsca dla polskich firm – zresztą niektóre już to dostrzegły.

– Czy także tych programistów, którzy przecież nie oferują produktów, chce pan zachęcić do włączenia się do PRO?

– PRO zdecydowanie jest odpowiednim miejscem dla takich osób i będą mile widziane w stowarzyszeniu. Problemy ochrony praw autorskich i własnej twórczości dotyczą przecież także twórców dokumentacji, projektów czy pomysłów, które materializują się w jakimś konkretnym rozwiązaniu. Jeżeli ktoś wykorzystując kilka produktów wykonał ciekawą aplikację, to podlega ona ochronie i może zostać zarejestrowana. Jeżeli ktoś buduje na przykład wyspecjalizowane systemy informacji geograficznej i komponuje je z różnych składników, nadając im pewną nadrzędną funkcję czy też dopasowuje je do specyfiki gminy, urzędu wojewódzkiego czy biura, to jest to właśnie produkt, który został twórczo wykonany i jak najbardziej podlega ochronie.

– W USA właściwie nie mówi się o programistach-twórcach, tylko o właścicielach praw autorskich. Paradoksalnie to nie autorów programów dotyczą prawa autorskie. Co pan o tym myśli?

– Przede wszystkim rozróżnienia wymagają prawa autorskie i majątkowe. Prawo majątkowe do oprogramowania przysługuje producentowi czy firmie, która je wyprodukowała. Większość z pracowników nie zdaje sobie sprawy z tego, że podpisując umowę o pracę rezygnuje z praw autorskich do tego, co wytworzy. Jeżeli chce się pewne prawa autorskie zastrzec dla siebie, należy uwzględnić to w umowie między pracownikiem a producentem. To się zdarza na razie w Polsce bardzo rzadko.

– A w USA się zdarza?

– Dotyczy to szczególnie środowisk silnych intelektualnie, np. uczelnianych, gdzie wchodzą w grę aspekty twórczości naukowej, czy nowych technologii na styku informatyki z produktami materialnymi. Natomiast uważam, że warto wszystkim uświadamiać, że oprogramowanie jest czymś, co podlega ochronie prawa autorskiego, nie tylko na tej zasadzie, że nie można kopiować gotowych produktów, ale już na etapie tworzenia. Każdy etap tworzenia oprogramowania tej ochronie podlega, łącznie z dokumentacją projektu i dokumentacją techniczną. To wszystko jest twórczością, wartością intelektualną, do której się prawo autorskie stosuje.

– Może więc zaistnieć taka sytuacja, że zdolny programista otrzymuje zadanie do wykonania i domaga się umowy z zastrzeżeniem prowizji od wielkości sprzedaży.

– Pracownik może sobie nawet zastrzec w umowie, w jakim procencie oprogramowanie należy do niego. Pytanie, czy firma taką umowę z pracownikiem podpisze, ponieważ nie leży to w interesie producenta. Jeżeli ktoś chce samodzielnie produkować oprogramowanie, to zakłada własną firmę.

– Wynika z tego, że samodzielne wykonanie programu jest najlepszym sposobem na zapewnienie sobie pełni praw autorskich?

– Tylko, że program napisany przez jedną osobę ma umiarkowane szanse konkurencji na rynkach światowych – to już nie te czasy. Oprogramowanie jest na ogół wynikiem pracy zespołowej, w której trudno zmierzyć ilość wkładu autorskiego poszczególnych programistów, dlatego bardzo trudno jest spisywać w takim przypadku umowy autorskie.

– Czy z tego powodu wyłączono w Art. 20 ustawy o prawach autorskich programistów z możliwości osiągania zysków?

– Ten artykuł miał stworzyć warunki do pokrycia strat autorów, jakie wynikają z tzw. indywidualnego użytku prywatnego. W odniesieniu do oprogramowania prawo do takiego użytku wykluczono innym artykułem ustawy.

– Czy jest prawdą, że jeżeli kupuje się jedną licencję, to program może być użytkowany legalnie tylko na jednym komputerze?

– Niekoniecznie, licencje bywają rozmaite. Licencja może być związana z użytkownikiem czy z liczbą jednocześnie pracujących użytkowników, licencja wielokrotna, licencja związana z komputerem, tzn. przypisana tylko i wyłącznie do tego komputera. Są programy, które np. numer seryjny ustalają na podstawie karty sieciowej do komputera. Ten przypadek, o którym pan mówi, jest częstszy. Jeżeli ktoś mimo choroby decyduje się pracować w domu i wtedy nie użytkuje tego programu w firmie, tylko w domu dokładnie w tym samym celu, to w większości przypadków postępuje zgodnie z licencją. Są też takie programy, w których licencja na to nie zezwala i wtedy stanowiłoby to naruszenie praw licencyjnych i mogłoby być ścigane.

– Czy w związku z rozwojem techniki może się zmienić polityka sprzedaży i licencjonowania oprogramowania? Na przykład czy zostanie złamany monopol dystrybutorów określonego producenta?

– To się zmienia już teraz. Trudno jest kontrolować bity przewożone na CD-romach przez granicę, a jeszcze trudniej dostrzec je, kiedy wędrują Internetem. Nowe systemy i techniki sprzedaży są zagrożeniem dla klasycznych firm dilerskich i dystrybucyjnych, które żyją z marży handlowej. Powstają alternatywne kanały sprzedaży, nie wymagające ich istnienia. Na pewno w rezultacie zmniejszą się ich obroty. Firmy dilerskie poszukują już teraz innych źródeł dochodów, a są nimi najczęściej konsultacje, szkolenia czy wdrożenia.

– Produkcja oprogramowania wymaga dużych nakładów i mało kto decyduje się finansować ją z kredytów przy obecnej stopie procentowej. Jak wygląda opieka nad tym ważnym pod względem gospodarczym rodzajem produkcji ze strony państwa, a w tym Departamentu Informatyki URM, KBN, Ministerstwa Finansów czy Ministerstwa Przemysłu i Handlu?

– Ta dziedzina nie ma w Polsce żadnych opiekunów. Są sporadyczne przypadki tworzenia oprogramowania na uczelniach, rzadko też zakończone sukcesem materialnym. Jeżeli już na jakiejś uczelni grupa zdolnych naukowców pracuje nad oprogramowaniem i dochodzi do zadowalających rezultatów, to zakłada poza uczelnią firmę, która doprowadza produkcję do finału. O takich przypadkach oczywiście słyszałem. Natomiast nie słyszałem, aby KBN finansował lub wspierał działalność firm software'owych, chyba że zajmują się one działalnością badawczą.

– Ale KBN wydaje opinie o innowacyjności produktu, co upoważnia do otrzymania zniżek podatkowych.

– Ja się z tym akurat nie zetknąłem. Jeżeli takie możliwości istnieją, to na pewno warto z nich skorzystać. Istotnie produkcja przemysłowa i tworzenie programów są przez instytucje finansowe traktowane podobnie. To jest jeden z ważniejszych powodów, dla których u nas w kraju trudniej zrobić program konkurencyjny na rynku światowym. Tylko dzięki dobrej organizacji i znalezieniu alternatywnych źródeł finansowania powstają produkty, które na rynku światowym zdobyły jakąś pozycję. Sądzę, że takie nieliczne sukcesy wynikają z dużych zdolności naszych informatyków i niezłego wykształcenia, a także dlatego, że wielu ludzi poświęca temu sporo czasu i energii. Na pewno nie stoi za tym jakiś potężny kapitał,

wzorowa organizacja czy dokładnie przemyślana strategia. Wynika to raczej z zamiłowania do spontanicznej działalności, silnej motywacji i ambicji. Programowanie potrzebuje słowiańskiego charakteru.
Spotkałem się niedawno z dawnymi moimi kolegami, tzw. „amerykanami". Oni bardzo szybko, po przyjeździe do Stanów byli w stanie udowodnić, że są znacznie lepsi od innych, dzięki temu robią tam karierę. Amerykanie dostrzegają ogromny potencjał nawet w małych firmach; uważają, że warto zainwestować w nie miliard dolarów i wówczas zaczynają stawiać ostre wymagania i narzucać pewien reżim. Tego źródła finansowego brakuje u nas, dlatego ten przemysł powstaje bardziej spontanicznie, a to trochę za mało.

– Kiedy oprogramowanie polskie było równie popularne jak zagraniczne, nie było chronione ustawą o prawie autorskim. Dopiero po upadku polskich firm software'owych i wejściu na nasz rynek oprogramowania zachodniego, zaczęto mówić o piractwie komputerowym. Być może dlatego produkcja oprogramowania nie rozwinęła się w Polsce, bo nie było dla niej normalnych szans rozwoju.

– W Polsce nigdy nie było takiej sytuacji, żeby można było legalnie używać cudzego oprogramowania, ponieważ można się było przed tym chronić korzystając z prawa cywilnego. Rzeczywiście wejście firm zachodnich, konieczność ochrony ich interesów i unormowania sytuacji prawnej oraz nasze ambicje europejskie zmobilizowały do wprowadzenia w życie tej ustawy. Regulacje prawne zostały wymuszone w pewnym stopniu przez polsko-amerykańskie kontrakty. Czy brak tego prawa utrudniał działalność twórczą polskich programistów? O wiele bardziej limitował ją brak perspektyw sprzedaży. Z własnych doświadczeń wiem, że nasza firma jeszcze na początku lat 90-tych sprzedawała oprogramowanie na zachód wyłącznie przez Szwecję. Prawie nie sprzedawaliśmy go w Polsce. Bez sprawnej telekomunikacji, bez porządnego systemu bankowego, łatwych przepływów gotówki, sprawnego fakturowania, czyli bez całej obudowy biznesowo-marketingowej nie można było zrobić dobrych interesów. Samo prawo jeszcze niczego nie zbuduje, ono stwarza sprzyjające warunki, ale nie są one wystarczające.

– Polityka rządowa, przez popieranie perspektywicznych kierunków działalności gospodarczej, także jest u nas niewystarczająca...

– Bez wątpienia informatyka, a w szczególności oprogramowanie, to dziedzina, w której można dużo zarobić, mimo silnej konkurencji. Co prawda inwestycyjnie jest kosztowna, ale nie niszczy środowiska, nie wymaga wielkich hal produkcyjnych, magazynów ani jakichś szczególnych materiałów – jest działalnością intelektualną o tzw. wysokim stopniu przetworzenia.

Są kraje, które starają się budować swój wizerunek eksponując duży potencjał informatyczny. Powstają w nich jak np. w Indiach „zagłębia programistyczne", które przyjmują każde zlecenie, robią to sprawnie, poprawnie i szybko, taniej niż Amerykanie. I to przynosi efekty. U nas nikt na taki pomysł nie wpadł.

– Ostatnio można zaobserwować podkupywanie zdolnych programistów z Filipin czy Indii do krajów o dużym potencjale przemysłu informatycznego. Także w Polsce np. kanadyjska ambasada pomagała w zorganizowaniu transferu naszych programistów. Co pan sądzi o tym drenażu mózgów?

– Autentyczny drenaż mózgów był na początku lat 90-tych. Wówczas istotnie wyprowadzono z Polski wielu niesłychanie zdolnych ludzi, których skusiły możliwości poprawy statusu ekonomicznego lub możliwość uczestnictwa w ważnym przedsięwzięciu. Teraz część tych ludzi wraca, a inni współpracują z polskimi firmami. Relacje polsko-zachodnie powoli się cywilizują. Przestają być relacjami informatycznego trzeciego świata z wielkimi potentatami.

– Dziękujemy za rozmowę.

Stowarzyszenie Polski Rynek Oprogramowania (PRO) prowadzi działalność służącą ochronie praw autorskich twórców oprogramowania, dokumentacji i rozwiązań informatycznych. W 1995 r. Minister Kultury i Sztuki udzielił zezwolenia Stowarzyszeniu PRO na zbiorowe zarządzanie prawami autorskimi do oprogramowania komputerowego. PRO prowadzi także Polski Rejestr Oprogramowania, który służy do ewidencji programów, baz danych, czcionek, grafiki komputerowej, utworów multimedialnych, itp. Dokonanie rejestracji oprogramowania następuje na pisemny wniosek właścicieli praw autorskich.

Adres: Warszawa, ul. Żurawia 4a, tel. 693-54-55.

POLIXEL

Dynamicznie działająca firma, rozwijająca się w dziedzinie komunikacji, poszukuje specjalistów informatyki, telekomunikacji oraz zarządzania i marketingu.

1. Dział Systemów Komunikacyjnych
opracowywanie projektów rozwiązań sieciowych LAN/MAN/WAN oraz zapewnienie wsparcia technicznego dla naszych partnerów i klientów. Kontakty z dostawcami zagranicznymi. (Ref. SK)

2. Logistyka
Zapewnienie sprawnej realizacji dostaw z importu i wysyłek do odbiorców. Kontakty i współpraca z dostawcami, odbiorcami, urzędami celnymi i przewoźnikami. (Ref. LG)

Firma umożliwia rozwój pracowników organizując różnorodne formy szkolenia. Wymagana znajomość angielskiego.

Zainteresowane osoby prosimy o przesłanie listu motywacyjnego oraz życiorysu zawodowego na adres firmy, z podaniem na kopercie symbolu referencji.

Polixel S.A., ul. Zakopiańska 6, PL 03-934 Warszawa

INFORMATYKA w dużych projektach informacyjnych

Jerzy Szyller

Redakcja miesięcznika INFORMATYKA już od ponad 30 lat bacznie przygląda się sukcesom, komentuje i sygnalizuje problemy pojawiające się w krajowej informatyce. INFORMATYKA stanowi wciąż trybunę, z której wypowiadają się na istotne tematy naukowcy, inżynierowie i użytkownicy systemów komputerowych. Jednakże dynamika rozwoju informatyki w Polsce jest coraz większa, zasięg jej oddziaływania gwałtownie rozszerza się na różne obszary życia codziennego, a systemy informatyczne stają się coraz bardziej dojrzałe. Redakcja miesięcznika pozostawiając w dalszym ciągu swoje łamy otwarte dla autorów ze środowiska akademickiego, pragnie też ukazywać „na gorąco" wszystko to, co dotyczy poważnych przedsięwzięć informatycznych. Mówiąc zaś o poważnych przedsięwzięciach mamy na myśli taką działalność, która przebiega przez wiele faz projektowania, jest realizowana przez dłuższy czas i angażuje duże zasoby ludzkie i finansowe.

Udanych przykładów takiej działalności mamy już w Polsce coraz więcej. Jednak wiele przedsięwzięć informatycznych wciąż kończy się fiaskiem, znacznym przekroczeniem terminów lub planowanych środków. Naturalnie INFORMATYKA chcąc przedstawiać te zakończone sukcesem, będzie również chętnie zamieszczać analizę projektów nieudanych, chętnie prezentowaną przez ich autorów. Im większa będzie u realizatorów świadomość o powodach fiaska ich przedsięwzięcia, tym łatwiej przyjdzie im unikanie błędów w przyszłości.

Zazwyczaj przedsięwzięcie informatyczne, w odniesieniu do przedsiębiorstwa lub organizacji przybierające różną postać, składa się z kilku faz. INFORMATYKĘ będą szczególnie interesowały cztery podstawowe fazy realizowania systemów informatycznych: definicji zakresu, wyboru dostawcy, wdrażania i faza użytkowej eksploatacji systemu informatycznego.

W pierwszej z tych faz projektanci muszą odpowiedzieć na pytanie jaki ma być cel i zakres planowanego przedsięwzięcia informatycznego. W tej – bodaj najważniejszej, bo decydującej o postaci całego przedsięwzięcia – fazie muszą brać udział specjaliści-konsultanci badający wiele aspektów działalności przedsiębiorstwa, od pozycji na rynku i obsługi klientów, przez produkcję, aż do finansów. W Polsce działa jeszcze stosunkowo mało firm zajmujących się doradztwem i konsultingiem tego typu. Są to przede wszystkim najbardziej znane, światowe firmy konsultingowe, działające w największych polskich i zagranicznych przedsiębiorstwach. Nie trzeba dodawać, że takie kompleksowe doradztwo, prowadzone przez renomowane firmy, jest bardzo kosztowne. INFORMATYKA będzie współpracowała ze wszystkimi firmami, które wyrażą ochotę przedstawienia na łamach naszego miesięcznika sposobu, w jaki realizują fazę definiowania zakresu zmian strukturalnych w organizacji klienta oraz zalecenia dla projektowanego systemu informatycznego. Będziemy również zachęcali nieliczne na razie firmy polskie do przedstawienia własnych osiągnięć na niwie konsultingu i doradztwa w zakresie informatyki oraz zarządzania. Takie informacje spotkają się z pewnością z dużym zainteresowaniem średnich i małych polskich przedsiębiorstw, które oczekują dobrej oferty, równocześnie o przystępnej cenie.

Kolejna faza realizacji przedsięwzięcia informatycznego, która bardzo interesuje redaktorów INFORMATYKI i jej czytelników, polega na prowadzeniu złożonego projektu systemu informatycznego według precyzyjnie sformułowanych zasad. Wiele projektów informatycznych nie udaje się, dlatego że mimo sformułowania dobrych założeń dla tworzonego systemu, w fazie definiowania zakresu realizacja przebiega chaotycznie, bez współpracy z najbardziej zainteresowanym, czyli klientem. Zazwyczaj sukcesem kończą się te przedsięwzięcia informatyczne, które są wdrażane według precyzyjnie skonstruowanego i przestrzeganego planu. Zaś wykonanie planu i jego kontrola są możliwe jedynie przy zastosowaniu odpowiedniej metodologii realizowania przedsięwzięć. Im większe przedsięwzięcie informatyczne, tym bardziej złożona i dojrzała powinna być stosowana metodologia. Firmy, które posiadają i stosują we wdrożeniach systemów odpowiednią i skuteczną metodologię, rozwijają i ulepszają ją latami. Dlatego najczęściej bardzo trudno skłonić je do ujawnienia tego, co stanowi najgłębszą tajemnicę ich skuteczności na rynku – metodologii.

INFORMATYKA stoi na stanowisku, że środowisko informatyczne w Polsce należy informować i edukować tak, aby możliwość popełniania błędów podczas realizacji dużych, wielomilionowych projektów informatycznych była coraz mniejsza. Dzisiaj prawie każda, nawet najmniejsza firma komputerowa mieni się integratorem, chcąc podnieść własny prestiż lub nie rozumiejąc co naprawdę oznacza ten termin. INFORMATYKA, wyjaśniając wiele nieporozumień w tej materii, będzie przedstawiała firmy, także krajowe, które dysponują wiarygodnymi technologiami informatycznymi, wdrażanymi według skutecznych metodologii.

Wreszcie ostatnia, najbardziej fascynująca faza, w której okazuje się, czy realizowane przez wiele miesięcy, a nawet lat, przedsięwzięcie informatyczne udało się. To znaczy, że organizacja i system informatyczny funkcjonują według uzgodnionych założeń. Pracownicy akceptują i wykorzystują w pełni funkcje systemu, co przynosi wymierne efekty ekonomiczne dla przedsiębiorstwa. W Polsce jest już wiele pozytywnych przykładów prawidłowo wdrożonych systemów informatycznych. INFORMATYKA będzie te przykłady pokazywać. Te przedsiębiorstwa i organizacje, które udostępnią nam informacje na temat własnych, działających bez zarzutu systemów informatycznych uwiarygodnią firmy komputerowe, z którymi współpracowały.

Duże przedsięwzięcia informatyczne kreują, tak jak każda dziedzina posiadająca własną specyfikę, określony, zazwyczaj hermetyczny język. W INFORMATYCE chcemy używać języka zrozumiałego dla wszystkich czytelników. Dlatego w „kąciku językowym" będziemy poruszać problemy związane z nowymi pojęciami, których ostatnio wiele pojawiło się w prasie fachowej. Tylko twórcza dyskusja może spowodować wyłonienie się i ujednolicenie nowego, polskiego nazewnictwa oraz wyeliminowanie wielu, często fatalnych, polskich odpowiedników dla nazw w języku angielskim. Na początek proponujemy *„reengineering"*.

Dużo mówiliśmy dotąd o przedsięwzięciach informatycznych. Intuicyjnie wyczuwamy o co chodzi. Jednakże dopiero lektura artykułu traktującego o systemie informatycznym jako przedsięwzięciu porządkuje wiedzę o tej tematyce. Artykuł pochodzi z ósmego numeru DEC-forum, ale po dwóch latach nic nie stracił z aktualności. Jego autor, Piotr Sobolewski jest wysokiej klasy specjalistą w zakresie realizacji i kierowania dużymi projektami informatycznymi. Zapraszamy do lektury.

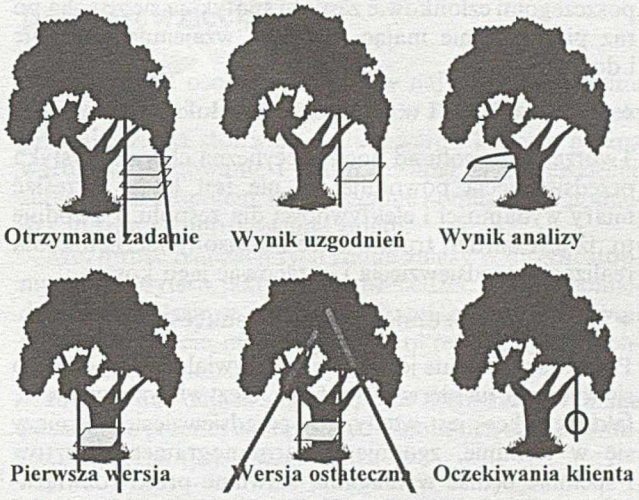
System informacyjny jako przedsięwzięcie

Piotr Sobolewski

Osiągnięcie sukcesu rynkowego przez przedsiębiorstwo zależne jest od szybkości i trafności podejmowanych decyzji. Informacja niezbędna dla podjęcia takiej decyzji dostarczana jest za pomocą współczesnych systemów informacyjnych. Systemy te, których tworzenie rozpoczęto ponad 30 lat temu, określane są też nazwą komputerowych systemów zarządzania informacją lub przetwarzania danych.

Stopień złożoności systemu informacyjnego jest wprost proporcjonalny do skali skomplikowania organizacji przedsiębiorstwa. Stworzenie systemu informacyjnego jest więc przedsięwzięciem na miarę stworzenia organizacji przedsiębiorstwa. Silna zależność pomiędzy systemem informacyjnym a prawidłowym działaniem przedsiębiorstwa oznacza konieczność takiej realizacji przedsięwzięcia, aby zakończyło się ono sukcesem.

Niniejszy artykuł ma na celu wprowadzenie czytelnika w podstawowe zagadnienia związane z realizacją przedsięwzięć, a w szczególności dotyczące kryteriów, jakimi mierzymy sukces oraz elementy niezbędne dla jego osiągnięcia. Pisząc o przedsięwzięciu mam na myśli przedsięwzięcie dotyczące realizacji informatycznego systemu informacyjnego. System taki może być tworzony od podstaw lub stanowić modyfikację już istniejącego. Informacje zawarte w artykule są nieformalnym opisem wiedzy i doświadczeń będących wynikiem lat analiz i pracy przy realizacji przedsięwzięć. Realizacja kolejnych zadań, napotkane błędy, porażki i sukcesy są źródłem doświadczeń, które prowadzą do wniosków, o których traktuje ten artykul. Tylko od nas zależy, czy będziemy wykorzystywać tę wiedzę, czy też sami dojdziemy do podobnych wniosków ucząc się na własnych błędach.



Rysunek 1. Czy tworzenie systemów informacyjnych zawsze musi tak wyglądać?

Przedsięwzięcie

Przedsięwzięcie – projekt lub program – można określić jako konkretną pracę charakteryzującą się dyskretnym czasem rozpoczęcia i zakończenia, której celem jest osiągnięcie wcześniej zdefiniowanych rezultatów. Podstawowe cechy typowego przedsięwzięcia to:

- **unikalność**

Wiekszość systemów tworzona jest tak, aby spełniać oczekiwania organizacji klienta. Jednak nie istnieje nic takiego, jak wzorzec organizacyjny przedsiebiorstwa, zatem nie ma też systemów, które dałoby się przenosić pomiędzy przedsiębiorstwami nie dokonując w nich jednocześnie znacznych zmian. Zwykle proces adaptowania gotowego systemu trwa latami.

- **ciągłość zmian**

Przedsiębiorstwo i jego organizacja podlegają ciągłym zmianom, które w konsekwencji wymuszają zmiany na systemie informacyjnym. Z drugiej strony następuje bardzo szybki rozwój technologii informatycznej, co pociąga za sobą zmiany systemu informacyjnego. Przedsięwzięcie trwające rok lub dłużej musi uwzględniać te zmiany.

- **udział różnych specjalistów**

Typowe przedsięwzięcie dotyczące tworzenia systemu informacyjnego wymaga zaangażowania ludzi o różnych specjalnościach. Tworzenie instalacji sprzętowej wymaga uczestnictwa specjalistów elektryków, czasami robot budowlanych i innego profesjonalnego wyposażenia. Samo oprogramowanie to nie tylko udział programistów, projektantów i analityków, ale także przedstawicieli użytkowników, którzy pomagają w tworzeniu specyfikacji i testowaniu systemu.

- **zespół ludzi, często tworzony jedynie na czas przedsięwzięcia**

Nawet duże organizacje, które zajmują się wyłącznie realizacją dużych przedsięwzięć dla swoich klientów, nie posiadają stałych zespołów do ich realizacji. Zespół tworzony jest dla realizacji konkretnego przedsięwzięcia, a poszczególne osoby są dobierane tak, aby w najwyższym stopniu spełniać wymagania stawiane na początku realizacji. Wadą takiego rozwiązania jest to, iż często poszczególni członkowie zespołu spotykają się ze sobą po raz pierwszy, nie mając czasu na wzajemne poznanie i dotarcie.

- **duże trudności w mierzeniu produktywności**

Tworzenie zespołu ad hoc i specyficzna charakterystyka przedsięwzięcia powodują, że nie jest łatwo określać miary wydajności i efektywności dla zespołu. Powoduje to, iż jest bardzo trudno określić zasoby niezbędne dla realizacji przedsięwzięcia i oszacować jego koszty.

- **odmienne rozumienie pojęcia sukcesu**

Pojęcie sukcesu nie jest wcale tak trywialne, jakby mogło się wydawać na pierwszy rzut oka. Oczywistym wydaje się fakt, że sukces jest wtedy, gdy przedsięwzięcie zakończy się w terminie, zgodnie z harmonogramem kosztów i spełniać będzie wymagania stawiane przed rozwiązaniem. Jednak czy naprawdę to jest najważniejsze dla klienta czy członków zespołu?

Miary sukcesu

Wymiernym testem – za którego pomocą określamy, czy tworzenie systemu informacyjnego zakończyło się sukcesem – jest stopień, w jakim spełnia on potrzeby organizacji (przedsiębiorstwa). Jest to także sposób, w jaki kierownictwo przedsiębiorstwa ocenia system informacyjny. System ten powinien długofalowo gwarantować przedsiębiorstwu osiąganie zysków, bez powodowania jakichkolwiek niepożądanych efektów ubocznych.

Na niższym szczeblu osoby zaangażowane w przedsięwzięcie mają dodatkowe miary sukcesu:

- Kierownictwo działu, które będzie bezpośrednio wykorzystywać wdrożony system, będzie oczekiwało maksymalnych korzyści dla swojego działu;
- Menedżer przedsięwzięcia będzie dokonywał oceny, biorąc pod uwagę różnice pomiędzy planowanym czasem i budżetem a osiągniętymi rezultatami;
- Zespół ludzi zaangażowanych w przedsiębiorstwie będzie brał pod uwagę oba poprzednie czynniki. Natomiast poszczególne osoby oczekują zadowolenia z pracy i praktycznych doświadczeń z uczestnictwa w przedsięwzięciu.

Elementy niezbędne dla osiągnięcia sukcesu

Analiza szerokiej gamy różnorakich przedsięwzięć daje możliwość określenia następujących czynników, od których zależy osiągnięcie sukcesu.

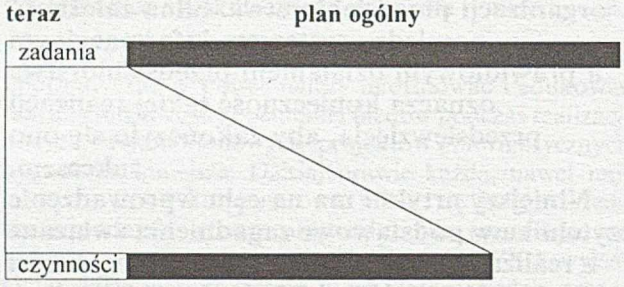
Określenie celów i uwarunkowań

Jasne sformułowanie celów i uwarunkowań wyspecyfikowane za pomocą zrozumiałych pojęć (tak aby były one przejrzyste, policzalne i mierzalne) jest pierwszym krokiem w realizacji przedsięwzięcia. Polega on na określeniu:

- celów, które powinny być mierzalne, realistyczne, zaakceptowane, sformalizowane i aktualne,
- zakresu definiującego granice przedsięwzięcia,
- uwarunkowań dotyczących czasu trwania, budżetu i wykorzystania zasobów.

Planowanie przedsięwzięcia

Ostrożne planowanie jest niezwykle ważne dla pomyślnego zakończenia przedsięwzięcia. Planowanie powinno przebiegać dwupoziomowo, z jednej strony na ogólnym poziomie dotyczyć całości przedsięwzięcia, a z drugiej w szczegółach określać poszczególne etapy. Plany powinny określać harmonogram odpowiedzialności oraz produkty dostarczane w poszczególnych etapach. Przyjęcie realistycznych planów pomaga w osiągnięciu celów, motywuje ludzi do pracy i zwiększa zaufanie do Menedżera przedsięwzięcia.



Rysunek 2. Planowanie dwupoziomowe.

Większość przedsięwzięć informatycznych jest niezwykle kosztowna i angażuje znaczne pieniądze oraz zasoby. Brak odpowiedniego planowania może doprowadzić do opóźnień, które natychmiast zwiększają koszt przedsięwzięcia. Dobre planowanie ma fundamentalne znaczenie dla ekonomii przedsięwzięcia, organizacji, dystrybucji zasobów i kontroli postępu prac. Zalety planowania są następujące:

- łatwość integracji poszczególnych akcji i ich unikalność;
- dokumentacja, którą można przeglądać i analizować w celu określenia, czy plan jest spójny i stabilny oraz czy spełnia stawiane przed nim wymagania;
- wczesne wykrycie potencjalnych problemów i możliwych źródeł opóźnień;

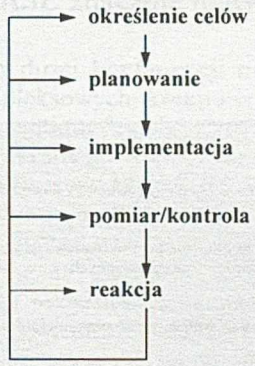
- możliwość koordynacji i kontroli przedsięwzięcia;
- ulepsza sam proces planowania w przyszłości. Rezultaty zebrane podczas planowania i realizacji przedsięwzięcia mogą zostać użyte w przyszłości do planowania innych przedsięwzięć.

Choć zalety planowania są bezdyskusyjne, to jednak posiada ono także pewne ograniczenia. Każda osoba tworząca plan musi zadać sobie na wstępie następujące pytania:

- jak daleko planować?
- jak szczegółowo planować?
- w jakim zakresie planować?
- jak dużo czasu poświęcić na planowanie?
- skąd czerpać dane?

Planowanie samo w sobie jest czynnością, która dzieli się na następujące działania:

- zdefiniowanie i akceptacja celów przedsięwzięcia oraz uwarunkowań,
- wyspecyfikowanie faz, zadań, działań i ich wzajemnych powiązań, niezbędnych dla osiągnięcia zamierzonych celów, w określonych uwarunkowaniach i przy założonym ryzyku,
- stworzenie wstępnego harmonogramu,
- określenie propozycji budżetu,
- określenie metod kontroli,
- specyfikacja i akceptacja zasobów i ich związków z przedsięwzięciem.



Rysunek 3. Współzależność planowania i mechanizmu kontroli.

Kontrola przedsięwzięcia

Realizacja przedsięwzięcia pozbawiona kontroli może doprowadzić do fiaska przedsięwzięcia, a przedsiębiorstwu przynieść znaczne straty. Kontrola przedsięwzięcia jest niezwykle ważnym elementem i sposób jej wdrożenia powinien zostać dokładnie zaplanowany.

Od momentu, kiedy plan zaczyna być realizowany, kontrola staje się niezbędna dla mierzenia postępu, identyfikowania odstępstw i przeprowadzania poprawek. Celem kontroli jest nie tylko identyfikacja błędów, ale także dostarczenie takich informacji, które zawrócą przedsięwzięcie na poprawne tory.

Mechanizm kontroli wymaga, aby podczas realizacji przedsięwzięcia zadane zostały następujące pytania:

- czy ciągle pracujemy w celu osiągnięcia poprawnych celów i czy uwarunkowania są ciągle te same?
- czy istnieją nowe, wymagające poprawek czy też niepotrzebne fazy zadania lub działania?
- czy harmonogram lub budżet nie powinny zostać skorygowane?
- czy nie należy poprawić specyfikacji zasobów i ich związków z przedsięwzięciem?

Mechanizm kontroli jest realizowany przez odpowiednio zaplanowane spotkania, przeglądy postępu prac, raporty, dokumentację kontrolną i nieformalne raporty.

Współuczestnictwo użytkowników

Stopień zaangażowania użytkowników determinuje osiągnięcie sukcesu. Specyfika przedsięwzięć dotyczących systemów informacyjnych jest taka, że częstokroć nie wystarczy służbowa zależność czy podległość. Krytycznym elementem staje się wtedy stopień, w jakim przedsięwzięcie i ludzie biorący w nim udział, są akceptowani przez przyszłych użytkowników.

Efektywność Menedżera przedsięwzięcia

Menedżer jest osobą, od której w największym stopniu zależny jest sukces przedsięwzięcia. Przed Menedżerem przedsięwzięcia stoi wiele zadań, wśród których można wyróżnić: planowanie i kontrolowanie przedsięwzięcia, zarządzanie zespołem, wspomaganie komunikacji pomiędzy ludźmi i kontrolowanie zmian.

Możliwość realizacji tych zadań zależy w dużym stopniu od autorytetu, jaki posiada Menedżer. Relacje pomiędzy Menedżerem a innymi osobami, zaangażowanymi w przedsięwzięcie, muszą być ściśle określone i powszechnie akceptowane.

Efektywna i dająca sukcesy praca Menedżera wymaga posiadania przez tę osobę wielu różnorakich cech charakteru. Do cech tych można zaliczyć:

- nastawienie na osiągnięcie sukcesu i pozytywne nastawienie do przedsięwzięcia;
- pracowitość i determinacja;
- wykształcenie i szybka przyswajalność wiedzy;
- silna osobowość i odporność na frustracje;
- zdyscyplinowanie.

Czynnik ludzki

Niezwykle ważnym elementem podczas realizacji przedsięwzięć informatycznych jest czynnik ludzki. W zasadzie zakończenie tego typu przedsięwzięcia jest w 100% zależne od ludzi, których praca jest w dużym stopniu twórcza. Niepowtarzalność przedsięwzięć i duży stopień komplikacji tworzonych systemów powodują, że czynnik ludzki jest niezwykle trudny do opisania i kontrolowania, jednak jego istnienie i rola nie podlega dyskusji. Wielokrotnie firmy posiadają doskonale wykształcone kadry, wspaniałe wyposażenie techniczne, a jednak pomimo to realizowane przez nie przedsięwzięcia kończą się fiaskiem. Po analizie okazuje się, że kierownictwo nie doceniło pewnych aspektów ludzkich.

Ludzie biorący udział w przedsięwzięciu mają różnorakie cechy charakteru. Niektóre osoby posiadają osobowość „kowboja", idącego przez życie przebojem i kierującego się wyłącznie własnym zdaniem, i tak też pracują. Inne osoby mają mentalność „mrówki" pracowicie i sys-

tematycznie wykonującej polecenia. W ramach realizacji większości przedsięwzięć niezbędne są osoby o obu tych charakterystykach. Podstawową sprawą jest tylko odpowiednie kierowanie zespołem złożonym z takich ludzi.

Podczas realizcji przedsięwzięcia członkowie zespołu podlegają różnym stresom. Stres związany jest zazwyczaj ze zbliżaniem się dat określających zakończenie etapu lub całości pracy. Stres ten jest naturalnym elementem każdego przedsięwzięcia, należy o nim pamiętać i starać się go redukować, o ile to możliwe. Należy jednocześnie za wszelką cenę unikać nowych stresów, wynikających z nierealistycznego planowania czy konfliktów międzyludzkich.

Sprawdzonym sposobem unikania problemów związanych z czynnikiem ludzkim jest dobra organizacja zespołu. Prawidłowa organizacja zespołu stawia przed nami następujące wyzwania:

- Taki dobór ludzi, aby stanowili zgrany zespół jednostek o odpowiednim wykształceniu i zaangażowanych w realizacje przedsięwzięcia;
- Takie określenie struktury zespoły, aby wzajemne relacje były ściśle zdefiniowane i nie stały się źródłem niepotrzebnych konfliktów. Każdy członek zespołu musi mieć określony zakres obowiązków i zakres odpowiedzialności;
- Przygotowanie odpowiedniego środowiska w postaci pomieszczeń do pracy i dyskusji, oraz odpowiednie wyposażenie techniczne (w pomieszczenium, w którym projektanci i programiści tworzą zręby systemu, nie może dzwonić jednocześnie kilka telefonów i odbywać się kilka ogólnosłyszalnych dyskusji);
- Określenie miar efektywności pracy członków zespołu, które zwiększają możliwość prawidłowego planowania i kontrolowania przebiegu przedsięwzięcia.

Może się wydawać, że w ocenie czynnika ludzkiego jesteśmy skazani jedynie na wyczucie czy własną subiektywną ocenę. Istnieją jednak techniki, które pomagają w analizie charakteru poszczególnych osób, jak i zachowania się całych grup. Techniki te są stosowane przy analizie zachowań ludzi pracujących razem przy realizcji przedsięwzięć. Nadrzędnym hasłem przy stosowaniu tych technik jest „nikt nie jest idealny – ale zespół może być".

LITERATURA

[1] Aoki Atsusi „Rafts for software engineers" – American Programmer, July 1993 Vol.6 No.7

[2] Constantine Larry L. „Coding cowboys and software sages" – American Programmer, July 1993 Vol.6 No.7

[3] Darnton G., Giacoletto S. „Information in the enterprise" – Digital Press 1992

[4] Project Management – Foundation I – materiały kursu Hoskyns Group plc.

Piotr Sobolewski jest pracownikiem Digital Equipment Polska.

Warunki prenumeraty na 1996 r.

Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

Zamawiający może otrzymywać zaprenumerowany przez siebie tytuł począwszy od następnego miesiąca po dokonaniu wpłaty. Zamówienia na zeszyty sprzed daty otrzymania wpłaty będą realizowane w miarę możliwości – z posiadanych zapasów magazynowych.

Warunkiem przyjęcia i realizacji zamówienia jest otrzymanie z banku potwierdzenia dokonania wpłaty przez prenumeratora. Dokument wpłaty jest równoznaczny ze złożeniem zamówienia.

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując środki pod adres:
Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-716 Warszawa, skr. poczt. 1004
konto:
PBK S.A. III 0/Warszawa nr 370015-1573-2720-3-67

Wpłaty na prenumeratę przyjmują także wszystkie urzędy pocztowe nadawczo-odbiorcze oraz doręczyciele na terenie całego kraju

Na blankiecie wpłaty należy czytelnie podać nazwę zamawianego czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty oraz własny adres.
Na życzenie prenumeratora, zgłoszone np. telefonicznie, Zakład Kolportażu ul. Bartycka 20, 00-950 Warszawa, (telefony: 40-30-86, 40-35-89 oraz 40-00-21 wew. 249, 293, 299) wysyła specjalne blankiety zamówień wraz z aktualną listą tytułów i cennikiem czasopism.

Odbiorcy zagraniczni mogą otrzymywać czasopisma poprzez prenumeratę dewizową (wpłata dokonywana poza granicami Polski w dewizach, wg cennika dewizowego z cenami podanymi w dolarach amerykańskich) lub poprzez zamówioną w kraju prenumeratę ze zleceniem wysyłki za granicę (zamawiający podaje dokładny adres odbiorcy za granicą, dokonując równocześnie wpłaty w wysokości dwukrotnie wyższej niż cena normalnej prenumeraty krajowej).

Egzemplarze archiwalne (sprzedaż przelewowa lub za zaliczeniem pocztowym) można zamawiać pisemnie, kierując zamówienia pod adresem: Wydawnictwo SIGMA NOT, Spółka z o.o. Zakład Kolportażu, 00-716 Warszawa, ul. Bartycka 20, paw. B, tel. 40-37-31, natomiast za gotówkę można je nabyć w Klubie Prasy Technicznej w Warszawie ul. Mazowieckiej 12, tel. 26-80-17.

Istnieje możliwość zaprenumerowania 1 egz. czasopisma po cenie ulgowej przez indywidualnych członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT oraz przez uczniów zawodowych i studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT lub szkoły.

W przypadku zmiany cen w okresie objętym prenumeratą Wydawnictwo zastrzega sobie prawo do wystąpienia o dopłatę różnicy cen oraz prawo do realizowania prenumeraty tylko w pełni opłaconej.
Cena jednego egzemplarza: normalna 5,00 zł (50 000 zł), ulgowa 3,75 zł (37 500 zł)
Wartość prenumeraty w zł:
Normalna: kwartalna 15,00 zł (150 000 zł), półroczna 30,00 zł (300 000 zł), roczna 60,00 zł (600 000 zł)
Ulgowa: kwartalna 11,25 zł (112 500 zł), półroczna 22,50 zł (225 000 zł), roczna 45,00 zł (450 000 zł)

publikacje

Wykorzystanie strukturalnych technik CASE w analizie systemów informacyjnych

Paweł Wiechecki
Piotr Kuboszek

Wdrażanie zintegrowanych systemów informatycznych w dużych organizacjach wymaga przeprowadzenia uprzedniej analizy. Jednym z najważniejszych warunków efektywnego jej wykonania jest zapewnienie dobrej komunikacji pomiędzy zespołem analityków a pracownikami firmy, w której odbywa się wdrożenie. Wykorzystanie narzędzi CASE znacznie ułatwia to zadanie.

W warunkach dużej konkurencji na polskim rynku dostawców kompleksowych, komputerowych systemów zarządzania, elementami decydującymi o wyborze przez użytkownika konkretnego rozwiązania są czas i koszty realizacji przedsięwzięcia. Wykorzystanie nowoczesnych technik programowania (4GL, RDBMS) pozwala maksymalnie skrócić czas wytwarzania oprogramowania. Największy wpływ na skrócenie czasu i ograniczenie kosztów realizacji przedsięwzięcia ma odejście od koncepcji tworzenia systemów dedykowanych na korzyść systemów powielanych, w dużym stopniu sparametryzowanych i umożliwiających ich zastosowanie u wielu użytkowników, reprezentujących rozmaite dziedziny gospodarki. Tworzenie systemów na potrzeby konkretnego klienta jest uzasadnione jedynie w przypadku rozwiązywania problemów specyficznych. Do obsługi typowych dziedzin działalności przedsiębiorstwa, regulowanych ogólnie obowiązującymi przepisami, stosowanie systemów powielanych jest uzasadnione rachunkiem ekonomicznym.

Cykl tworzenia systemu powielanego u użytkownika składa się z następujących etapów:

- faza negocjacyjna: prezentacje handlowe, które powinny odpowiedzieć na pytanie, czy system spełnia kluczowe potrzeby danego klienta,
- uzgodnienie warunków kontraktu,
- określenie wstępnego harmonogramu działań implementacyjnych,
- analiza systemu informacyjnego użytkownika,
- sformułowanie i analiza szczegółowych potrzeb użytkownika,
- instalacja sprzętu i oprogramowania,
- adaptacje i modyfikacje oprogramowania,
- wdrażanie systemu,
- eksploatacja i utrzymanie systemu,
- wycofanie systemu z eksploatacji.

Mimo, że w ciągu ostatnich kilku lat obserwuje się, że dostawcy dużych pakietów zarządzania kładą nacisk na fazę wdrażania, na skutek niedoceniania fazy wstępnej analizy nie zawsze realizacja projektu kończy się sukcesem. Celem analizy jest przede wszystkim zebranie informacji niezbędnych do optymalnego skonfigurowanie parametrów systemu, co umożliwi pełne wykorzystanie jego możliwości. Zmiana w trakcie procesu wdrażania lub eksploatacji niektórych kluczowych konstrukcji jest bardzo utrudniona, a czasami wręcz niemożliwa do wykonania. W związku z powyższym coraz bardziej zwraca się uwagę na korzyści płynące z pracy analityków, których zadaniem jest przygotowanie użytkownika oraz pomoc służbom wdrażającym system. Z uwagi na powtarzalny charakter cyklu życia systemu u wielu użytkowników w praktyce mamy do czynienia z modelem zaproponowanym przez Fry'ego.

Podstawowym zadaniem analizy jest identyfikacja danych przetwarzanych w systemie, ich przepływów oraz wykorzystywania w procesach, wyszczególnienie obiektów systemu, ich wzajemnych związków oraz danych je opisujących.

Analiza przedwdrożeniowa składa się z następujących elementów:

- identyfikacja oraz ocena funkcjonującej struktury organizacyjnej,
- analiza obiegu dokumentów,
- rozpoznanie dotychczas wykorzystywanych systemów informatycznych,
- ocena możliwości konwersji danych oraz wykorzystania posiadanego już sprzętu,
- określenie granic systemu informatycznego,
- określenie przepływów informacji zarówno wewnątrz systemu, jak i komunikacja procesów z obiektami zewnętrznymi,
- ocena przygotowania użytkownika do pracy z nowym systemem,
- utworzenie docelowego modelu przepływów informacyjnych,
- uzgodnienie ewentualnych zmian w organizacji pracy

oraz zakresie obowiązków poszczególnych służb i stanowisk pracy,
- szczegółowe rozpoznanie oraz sformułowanie potrzeb użytkownika nie zaspokajanych przez system powielany,
- ujawnienie potencjalnych trudności mogących wystąpić w trakcie wdrożenia,
- określenie najistotniejszych parametrów systemu,
- wspomaganie użytkownika przy podejmowaniu strategicznych decyzji.

W przypadku stosowania tradycyjnych metod, spośród wymienionych funkcji najbardziej uciążliwe jest opisanie procesów, przepływów danych, repozytoriów, obiektów zewnętrznych i ich wzajemnych interakcji. Opis słowny, w przeciwieństwie do graficznych form prezentacji tych zjawisk, jest nie tylko pracochłonny, ale także mało czytelny. Opis słowny utrudnia weryfikację opracowanego projektu przez użytkownika oraz efektywną komunikację pomiędzy analitykiem a przedstawicielami użytkownika w trakcie określenia założeń i wdrożenia systemu. Szczególnie dużego znaczenia nabiera możliwość wykorzystania diagramów przepływów danych. Wykonanie tych zadań może być wspomagane przez zastosowanie technik CASE.

CASE jest akronimem słów Computer Aided Software Engineering, co w polskim tłumaczeniu brzmi: Komputerowo Wspomagana Inżynieria Oprogramowania. CASE jest relatywnie nową technologią, powstałą w wyniku zapotrzebowania na usprawnienie procesu tworzenia oprogramowania i zautomatyzowania metod strukturalnej analizy, projektowania i modelowania danych. Stworzenie i popularyzacja takiego narzędzia, jakim jest CASE, były możliwe dzięki szybkiemu rozpowszechnieniu się stosunkowo tanich, wydajnych komputerów osobistych.

CASE jest terminem używanym do opisu środowiska zintegrowanych narzędzi automatyzujących proces rozwoju oprogramowania strukturalnego i narzucających rygor i dyscyplinę działaniom, związanym z inżynierią oprogramowania. Istnieje wiele metod korzystania z narzędzi CASE. Również producenci oprogramowania dostarczają różnych narzędzi CASE, które pokrywają jeden lub więcej etapów, takich jak: analiza wstępna, projektowanie, programowanie oraz testowanie.

Należy pamiętać, że narzędzia CASE nie są magicznym rozwiązaniem, ale dostarczają środków umożliwiających automatyzację pewnych etapów tworzenia systemów. Aby móc posługiwać się efektywnie omawianymi narzędziami, należy zrozumieć ustrukturyzowane metody opracowania systemu, ich zalety, wady i ograniczenia. Projekt systemu, który bazuje na błędnych lub nieprecyzyjnych założeniach, nie doprowadzi do oczekiwanego rezultatu bez względu na to, czy będzie tworzony przy użyciu tradycyjnych metod czy też z użyciem technik CASE. Innymi słowy, jeśli nie wiadomo jak ma wyglądać system (uwzględniając budżet, harmonogram, jakość, wymagania użytkowników, itp.) bez użycia CASE, wykorzystanie narzędzi wspomagających tworzenie oprogramowania nie rozwiąże tego problemu.

W naszej firmie najczęściej stosowanym narzędziem CASE jest oparty na składnicach danych EasyCASE, który wykorzystuje następujące techniki:
- analizę strukturalną,
- projektowanie strukturalne,
- modelowanie danych i informacji.

Wspomaganie odbywa się m. in. przez udostępnienie różnorodnych typów diagramów – graficznej reprezentacji struktury, która jest przedmiotem zainteresowania twórców systemu. Do najważniejszych typów diagramów oferowanych przez omawiane narzędzie możemy zaliczyć:
- diagramy przepływu danych – data flow diagrams (DFD)
- diagramy związków obiektów – entity relationship diagrams (ERD)
- diagramy życia obiektów – entity life history diagrams (ELH)
- diagramy struktury danych – data structure diagrams (DSD)
- schematy strukturalne – structure charts (STC)

EasyCASE dostarcza metodyk używanych w procesie analizy i projektowania systemu. Są one oparte na koncepcji budowy systemów ustrukturyzowanych. Wykorzystują diagramy w celu graficznej reprezentacji komponentów, relacji pomiędzy komponentami oraz stanów systemu. Umiejscowienie na diagramach opisów tekstowych oraz definicji składników graficznych w składnicach danych uzupełnia informacje o modelu systemu.

Spośród wielu dostępnych w EasyCASE metodyk analizy strukturalnej (takich jak np.: SSADM, Yourdon & DeMarco, Chen, Martin, etc.) najczęściej stosujemy metodykę Gane & Sarson.

Symbolika diagramów DFD, które są przez nas głównie wykorzystywane, ogranicza się do kilku elementów, które zostały przedstawione w książce P. Fuglewicza, K. Stąpora i A. Trojnara „CASE dla ludzi".

W przypadku analizy przedwdrożeniowej systemu powielanego wykorzystanie diagramów ERD, będących z założenia kolejnym etapem analizy, jest mało przydatne, ponieważ doświadczenia wykazały, że modele te są trudne do zrozumienia przez osoby nie posiadające przygotowania informatycznego i w związku z tym weryfikacja ich przez użytkownika najczęściej nie wnosi żadnych wartościowych spostrzeżeń.

Diagramy DFD są sporządzane na co najmniej trzech poziomach. Hierarchicznie najwyższym diagramem jest tzw. diagram kontekstowy.
Musi on zawierać następujące elementy:
- jeden i tylko jeden proces obejmujący cały system informatyczny,
- wszystkie istotne obiekty zewnętrzne oddziałujące na system,
- przepływy informacji wychodzące z systemu do obiektów zewnętrznych i odwrotnie.

Na podstawie diagramu kontekstowego określamy, które elementy systemu informacyjnego wchodzą w granice systemu informatycznego, a które pozostają poza nim.

Kolejny poziom stanowi DFD poziomu zerowego. Przedstawia on podstawowe procesy funkcjonujące w ramach systemu informatycznego oraz przepływy informacji między procesami ze szczególnym uwzględnieniem procesów odpowiedzialnych za komunikację z obiektami zewnętrznymi.

Następne poziomy przedstawiają strukturę przepływu informacji w sposób bardziej szczegółowy i opisują poszczególne obszary systemu zarządzania. Ilość tych poziomów zależy od stopnia złożoności opisywanego podmiotu. W przeciwieństwie do poprzednio wymienionych poziomów, DFD poziomu pierwszego i niższych najczęściej składa się z kilku osobnych diagramów opisujących określone agendy.

dokończenie na s. 31 →

Relacyjne a obiektowe bazy danych - rywalizacja czy koegzystencja?

Witold Abramowicz
Grzegorz Bartoszewicz
Leszek A. Maciaszek

Od około 1985 roku na rynku zaczęły dominować relacyjne systemy zarządzania bazami danych. Ta dominacja jest bezdyskusyjna na polu aplikacji użytkowanych na komputerach osobistych i systemach Unix. Jak pokazują statystyki ciągle ponad 50% zastosowań dużych baz danych na komputerach typu mainframe nie są bazami relacyjnymi (np. IMS, IDMS, TOTAL, ADABAS). Obiektowe bazy danych znalazły wiele zastosowań komercyjnych dopiero w ostatnich kilku latach i to tylko w wybranych obszarach takich, jak np. systemy informacji geograficznej.

Znaczący jest również fakt, że roczna wartość obrotów na rynku baz obiektowych jest obecnie znacznie mniejsza niż wartość obrotów na rynku baz relacyjnych. Mimo to bazy obiektowe wywołują coraz szersze zainteresowanie zarówno wśród dostawców systemów informacyjnych, jak i potencjalnych użytkowników. Do najpopularniejszych relacyjnych systemów zarządzania bazami danych (SZBD) na platformie Unix można zaliczyć: ORACLE, SYBASE, INGRES i INFORMIX oraz DB2 dla baz gigabajtowych. Większość obiektowych SZBD była przeznaczona dla systemu Unix. Większość systemów obiektowych powstała w oparciu o język C++, chociaż szybko rośnie liczba takich systemów zbudowanych z wykorzystaniem języka Smalltalk. Do najczęściej stosowanych systemów obiektowych baz danych należą ObjectStore, VERSANT, Objectivity/DB, Gemstone, ONTOS, O_2, POET, MATISEE, ITASCA. Rosnącą popularnością cieszą się systemy hybrydowe (obiektowo-relacyjne) takie, jak np. UniSQL (relacyjny klient korzystający z obiektowego serwera), Illustra, HP Odapter (dostarcza obiektowy interfejs do relacyjnego serwera).

Systemy relacyjne są używane zazwyczaj w systemach informacyjnych zarządzania, natomiast bazy obiektowe są przeznaczone dla nowoczesnych zastosowań: w pracy grupowej (wspomaganie projektowania i wytwarzania – CAD/CAM, współtworzenie i publikowanie, automatyzacja prac biurowych), aplikacjach naukowych i medycznych, systemach czasu rzeczywistego i multimedialnych.

Mogą być one postrzegane jako naturalne rozszerzenie obiektowych języków programowania i dlatego są atrakcyjne dla programistów pracujących z C++ i Smalltalk. Mogą być także użyteczne dla integracji oprogramowania pochodzącego z różnych źródeł z gotowymi aplikacjami. Są tak projektowane, aby efektywnie wykorzystać nowoczesną architekturę stacji roboczych.

Cechy obiektowej bazy

Obiektowa baza danych jest oczywiście bazą danych i dlatego powinna posiadać przynajmniej następujące cechy:

- udostępniać trwałą pamięć o dowolnym rozmiarze,
- zarządzać transakcjami w warunkach współbieżności,
- zarządzać transakcjami umożliwiając odtwarzanie danych,
- umożliwić autoryzację dostępu i bezpieczeństwo danych,
- zapewnić spójność i integralność danych,
- umożliwiać formułowanie zapytań ad hoc (język zapytań),
- zapewnić logiczną i fizyczną niezależność danych,
- udostępnić generator aplikacji,
- zapewnić obsługę rozproszonej bazy danych o architekturze klient-serwer.

Omówimy te własności porównując ich charakterystyki dla baz relacyjnych i obiektowych.

Trwała pamięć o dowolnym rozmiarze

Baza danych obsługuje trzy rodzaje danych: przejściowe – te, które są dostępne tylko podczas działania pojedynczego programu, stabilne – możliwe do odzyskania w przypadku błędu występującego w trakcie realizacji procesu oraz trwałe – dostępne także poza procesem, który je stworzył.

W relacyjnych bazach danych trwałymi są wszystkie dane pamiętane w tabelach. Operacje SQL uzyskują automatyczny dostęp do tych danych. W bazie danych są zapisane wszystkie wprowadzane zmiany. Natomiast w bazach obiektowych obiekty mogą być w trakcie ich powoływania określane jako przejściowe lub trwałe (np. nowy operator C++). W niektórych systemach istnieje możliwość zmiany określenia po zadeklarowaniu obiektu.

Zarządzanie transakcjami w warunkach współbieżności

Transakcja bazy danych jest zbiorem akcji wykonanych w bazie, które przenoszą ją z jednego dopuszczalnego stanu do stanu innego. Transakcje w systemach relacyjnych są najczęściej krótkie. System stara się je zrealizować w najkrótszym możliwym czasie, tak aby zasoby bazy danych mogły być zwolnione dla dokonania następnej

transakcji współbieżnej. Spełniają zatem *własności ACID* – Atomicity (atomizacji), Consistency (spójności), Isolation (wydzielenia) i Durability (trwałości).

Obiektowe bazy danych mają wiele innych mechanizmów ułatwiających obsługę transakcji współbieżnych, które nie są możliwe w systemach relacyjnych. Należą do nich:

- możliwość kolejkowania dla krótkich blokad,
- informowanie o dostępnych blokadach,
- stałe blokady dla zarządzania współbieżnością podczas długich transakcji,
- wersje obiektów,
- ułatwienia dla wyłączania blokad i ich dopasowania do indywidualnych potrzeb,
- możliwość pracy w sesji nie zakładającej konieczności blokowania z oznaczonymi czasowo obiektami,
- transakcje zagnieżdżone.

Zarządzanie transakcjami umożliwiające odtwarzania danych

Oprócz współbieżności, system zarządzania transakcjami obsługuje również zadania związane z odzyskiwaniem informacji. Do podstawowych mechanizmów służących do powrotu do stanu przed awarią (tzw. podniesienie bazy danych) należą: rejestrowanie danych w czasie pracy (dziennik transakcji) i archiwizowanie danych. Wspólnie mechanizmy te gwarantują odzyskiwanie danych przez mechanizmy wycofania i przesyłania transakcji. Odzyskiwanie przez wycofanie zapewnia, że nie potwierdzone transakcje zostaną usunięte z bazy. Odzyskiwanie przez przesyłanie zapewnia natomiast możliwość potwierdzania przez użytkownika zapisu transakcji do bazy. W rozproszonych systemach klasy klient/serwer dla odzyskiwania danych po awarii, systemy zarządzania bazą danych używają potwierdzania dwufazowego lub replikacji.

Relacyjne i obiektowe bazy danych używają zasadniczo tych samych technik odtwarzania, lecz implementacje relacyjne są solidniejsze i pewniejsze. Dla przykładu, dynamiczne archiwizowanie jest czymś normalnym dla systemów relacyjnych, natomiast niemożliwe w systemach obiektowych. Z drugiej strony, obiektowe bazy danych zezwalają na wyłączenie prowadzenia dziennika transakcji w celu zwiększenia efektywności działania aplikacji, w zastosowaniach, w których odzyskiwanie danych nie jest najważniejszą sprawą. Wyłączenie śledzenia w bazach obiektowych może mieć miejsce w przypadku przesyłania danych z grupowej do prywatnej bazy danych. Wynika to z faktu, że oryginały kontrolowanych obiektów pozostają bezpieczne w grupowej bazie danych.

Autoryzacja i bezpieczeństwo

Od obiektowych systemów zarządzania bazą danych oczekuje się nie tylko prostych mechanizmów autoryzacji i ochrony. Muszą one być rozszerzone w celu uwzględnienia złożoności struktury obiektów. Dla przykładu w bazach relacyjnych nie określa się przywilejów dla pojedynczych elementów tabeli, leżących na przecięciu wiersza i kolumny. Taka komórka jest dopuszczalnym obiektem w obiektowej bazie danych i musi podlegać ochronie. Podobnie autoryzacji wymaga partycypacja obiektu w klasie dziedziczenia lub klasie kompozycji. Obecnie, przy dostępie do obiektowej bazy danych, jest możliwa ochrona przez sprawdzenie autentyczności identyfikatora użytkownika. Dodatkowa autoryzacja może być uzyskana przez napisanie procedury kontroli dostępu.

Spójność i integralność

Spójność i integralność relacyjnych baz danych jest z reguły uzyskiwana przez implementację integralności referencyjnej pomiędzy tabelami. W tym celu systemy relacyjne wyposażają projektantów w nie wykluczający się wybór pomiędzy deklaratywną integralnością referencyjną w wyrażeniu CREATE TABLE (w DB2) a rozwiązaniami proceduralnymi opartymi na wyzwalaczach (trigger).

Obiektowe bazy danych oferują rozwiązania deklaratywne i proceduralne dla zapewnienia spójności i integralności. Różnice zaznaczają się w innej implementacji związków między danymi: w połączeniu z wartościami kluczy głównych i obcych w bazach relacyjnych i w bezpośrednim łączeniu obiektów używających identyfikatorów obiektowych (IO-s) w bazie obiektowej. Oczywiście obiektowa baza danych może również wiązać obiekty przez ich wartości, lecz jest to nieefektywne w porównaniu z wykorzystaniem IO-s.

Dodatkowa ochrona spójności i integralności obiektowej bazy danych – podobnej do opcji ON DELETE w definicji tabeli relacyjnej – musi być kodowana przez programistę. Można oczekiwać szybkiego postępu w tym zakresie. Wiele z proponowanych rozwiązań było już dostępne w systemach przedrelacyjnych (np. IDMS, RODAN), gdzie związki między danymi były implementowane przez łączniki. Integralność referencyjna była lepiej wspomagana niż we współczesnych relacyjnych bazach danych.

Język zapytań

Język zapytań jest nieprecyzyjnym terminem używanym dla określenia języka manipulacji danymi. Termin ten został spopularyzowany przez relacyjny język zapytań – SQL. Jest wiele wariantów SQL-a: interaktywny SQL, wbudowany SQL, 4GL-SQL, proceduralne rozszerzenia do interaktywnego SQL. SQL jest używany do zapytań i aktualizacji tabel. Spełnia własność domknięcia, tzn. bierze tablicę na wejściu i zwraca inną tablicę na wyjściu. Ocena zapytania jest oparta na porównaniu wartości atrybutów (kolumn) umieszczonych w warunkach szukania (predykat zapytania) z elementami bazy danych.

Interaktywny SQL używa tablic dokładnie w czasie dostępu do bazy danych i z tego powodu może stać się niebezpieczny w rękach niedoświadczonych użytkowników. W praktyce nie jest on oferowany użytkownikom jako język wolnej manipulacji zawartości bazy danych. Zamiast tego użytkownikom dostarczany jest interfejs do bazy danych, który jest programowany przy użyciu wbudowanego SQL lub 4GL-SQL. To niedopasowanie jest niedogodnością takiego podejścia. SQL jest częścią innego języka. System utrzymuje dwa typy systemów i dwa środowiska pracy. Nawet jeżeli SQL jest kompletny od strony przetwarzania (tak jak w różnych jego proceduralnych rozszerzeniach – SYBASE Transact SQL lub ORACLE PL/SQL), to odczuwa się niedostatek narzędzi

zasobowych. Na przykład nie można wyświetlać grafiki w oknach aplikacji.
W architekturze klient/serwer, zapytanie SQL jest zawsze wykonywane na serwerze bazy danych, ponieważ tylko baza na serwerze zna definicję tablic i inne parametry niezbędne do optymalizacji zapytania. Dla typowej aplikacji biznesowej, realizacja zapytania na serwerze jest efektywniejsza.

Większość programów dla obiektowej bazy danych jest napisana w języku ściśle zintegrowanym z jednym z języków obiektowych, najczęściej C++ lub Smalltalk. Takie oprogramowanie korzysta ze schematu bazy danych i jest używane do przetwarzania transakcji. Jednak użytkownicy bazy obiektowej odczuwają także potrzebę bardziej bezpośredniego dostępu do bazy, przy użyciu języka zapytań. Dla zaspokojenia tych potrzeb, bazy obiektowe dostarczają narzędzi typu ObjectSQL lub OSQL – rozszerzony SQL do pracy ze strukturami obiektowym.

Przy założeniu, że powyższe trudności implementacyjne zostały pokonane, ObjectSQL ma wiele udogodnień zwiększających jego sprawność w stosunku do relacyjnych SQL takich, jak: przetwarzanie przez połączenia, eliminacja niezgodności impedancyjnej (impedance mismatch), możliwość oceny identyfikatorów i połączeń obiektowych po stronie klienta, a następnie odzyskiwania obiektów z bazy danych po stronie serwera (jeśli to potrzebne, te identyfikatory obiektowe i połączenia są odnajdywane przez wcześniejsze przeszukania na wartościach obiektów).

Logiczna i fizyczna niezależność danych

Niezależność danych określa stopień, do którego zmiany w logicznej (niezależność logiczna) i fizycznej strukturze danych (niezależność fizyczna) nie są uzależnione od istniejących programów (tzn. programy nie muszą być modyfikowane lub nawet rekompilowane). Oparte na modelu logicznym relacyjne bazy danych wykazują wysoki stopień fizycznej niezależności.

Logiczna niezależność danych jest zatem stosunkowo silna w modelu relacyjnym, szczególnie w odniesieniu do rozszerzenia schematu i dostarczenia programistom odpowiadających im dobrych stylów programowania.
W bazach obiektowych bardzo dobrym wspomaganiem logicznej niezależności danych jest hermetyzacja danych przez metody. Niestety w praktyce hermetyzacja może być wynikiem kompromisu pomiędzy mechanizmami języka zapytań a dziedziczeniem. Jego rozwiązanie ma miejsce w konkretnej aplikacji.

Generowanie aplikacji

Programiści relacyjnych baz danych używają do kodowania generatorów aplikacji (języki czwartej generacji – 4GL). Tylko sporadycznie i w specjalnych zastosowaniach używany jest interfejs programowy do języków 3GL. Nie jest zatem niespodzianką, że programiści baz obiektowych również oczekują narzędzi klasy 4GL. Takie narzędzia zaczynają być oferowane (np. VERSANT ARGOS). Nie dorównują one jeszcze relacyjnym 4GL. Tak można tłumaczyć rosnącą popularność języka Smalltalk. Środowiska Smalltalk dają możliwości prototypowania. Wygrywają z C++, jako platforma 4GL.

Rozproszone bazy danych typu klient/serwer

Dostęp do współczesnych baz danych odbywa się przez stacje robocze. Są one często konfigurowane w architekturze klient/serwer. W tym modelu aplikacja jest klientem i jest realizowana na stacji roboczej. Baza danych, która dostarcza dane do aplikacji, jest serwerem (w rozumieniu programowym). Węzeł sieci z bazą danych może być w niektórych transakcjach klientem, a w innych serwerem. W rzeczywistych rozproszonych bazach danych, różnica pomiędzy klientem a serwerem jest czysto logiczna (teoretyczna) i klient może uzyskiwać dostęp do wielu serwerów jednocześnie.
Klient i serwer mogą być na tym samym lub na różnych komputerach. Jeśli są na różnych komputerach, wtedy klient i serwer wykonują dwa różne procesy. Jeśli są na tym samym komputerze, wtedy system zarządzania bazą ma możliwość pracy klienta i serwera w tym samym procesie.

Należy zwrócić uwagę na ważną różnicę pomiędzy realizacją modelu klient/serwer w systemie relacyjnym i obiektowym. W bazie relacyjnej zapytania są zawsze realizowane na serwerze. Zwiększa to wydajność typowych transakcji biznesowych, przetwarzających duże zbiory danych po to, by określić krótkie w porównaniu z nimi odpowiedzi. W nowoczesnych bazach obiektowych zapytania mogą być realizowane zarówno na kliencie, jak i serwerze. Obliczenia realizowane po stronie klienta są efektywniejsze w większości zastosowań w systemach pracy grupowej. Aplikacje kontrolują wówczas w długiej transakcji istotne fragmenty bazy przesyłane z serwera do klienta. Ponieważ bazy relacyjne nie wspomagają grupowego przetwarzania aplikacji, nie implementuje się w nich możliwości wykonywania zapytań na kliencie.
Wiele innych czynników może służyć do oceny rozproszonych baz danych. Dostawcy zarówno systemów relacyjnych, jak i obiektowych starają się przełamać ograniczenia technologiczne. Konkurencja jest ostra, lecz wiele spośród zapowiedzi dostawców powinno być traktowane z dużą ostrożnością. Nie ulega wątpliwości, że zmniejsza się technologiczna luka pomiędzy rozwiązaniami w obszarze systemów relacyjnych i obiektowych. Poniżej zestawiono na pewno niekompletny wykaz aktualnie rozwiązywanych problemów baz rozproszonych:

- dwufazowe potwierdzanie transakcji (niektórzy dostawcy obiecują tu wspomaganie, a następnie zlecają ... programistom ich implementację);
- replikacje (niektórzy dostawcy idą tak daleko, że dostarczają niezależny serwer replikacji, podczas gdy inni czynią użytkowników odpowiedzialnymi za spójność replikacji);
- projektowanie wielowątkowe – wielostrumieniowe (nowoczesne bazy danych wykorzystują wiele strumieni w pojedynczym procesie dla uzyskania dużej wydajności; niektóre obiekty bazy używają mniej efektywnych technik, tzn. oddzielny proces dla każdego użytkownika lub pojedynczy proces dla obsługi wszystkich żądań użytkowników);
- rozproszone aktualizacje (większość systemów mocno ogranicza rozproszone aktualizacje i nie pozwala na pojedynczą aktualizację w wielodostępnej bazie; użytkownicy mogą obsługiwać takie aktualizacje etapami);
- rozproszona optymalizacja zapytań (nie jest niespo-

dzianką, że systemy relacyjne oferują tutaj znacznie bardziej przejrzyste i efektywne algorytmy);
- migracja danych (obiektowe bazy danych mają tutaj przewagę, ponieważ obiekty mają przypisane logiczne identyfikatory, które są niezależnie tłumaczone, przynajmniej w teorii, na podstawie fizycznego położenia obiektu);
- różnorodność (obiektowe bazy danych mają przewagę, ponieważ technologia obiektowa ujednolica przetwarzanie obiektów przez publiczny interfejs, który hermetyzuje implementacje i różnice platform).

Co stanowi o tym, że baza jest obiektowa?

Obiektowa baza danych różni się od relacyjnej wieloma cechami. Do najważniejszych należą: zgodność obiektów, hermetyzacja, generalizacja i dziedziczenie, agregacja i wspomaganie multimedialne, długie transakcje, wersje i konfiguracje, sprawność. Pokrótce zostaną tu one omówione.

Zgodność obiektowa

Koncepcja zgodności obiektowej wyjaśnia olbrzymią lukę pomiędzy technologią obiektową i relacyjną, która będzie trudna do zniwelowania przez relacyjne bazy danych. Przez „niwelację" rozumiemy tutaj wprowadzenie rozszerzeń w architekturze technologii relacyjnej, a nie symulowanie koncepcji.

W obiektowej bazie danych generowany jest przez system unikalny dla całej bazy danych *identyfikator obiektu* (IO), który jest całkowicie niezależny od stanu obiektu lub jego adresu.

Wszystkie obiekty w bazie obiektowej są pamiętane wraz z IO. Obiekty powiązane są ze sobą z wykorzystaniem IO. Programy identyfikują obiekty dzięki IO. Identyfikatory IO umożliwiają stosowanie najbardziej efektywnych sposobów przetwarzania danych. Kiedy obiekty są sprowadzane do pamięci w celu przetwarzania, IO są automatycznie dodawane do adresów pamięci i są w ten sposób natychmiast dostępne dla programów. Oczywiście dla przetwarzania zapytań, obiektowe bazy danych posiadają również możliwość dostępu do danych przez klucze obiektów. Jednak pomimo tego większość operacji połączenia może być pominięta, ponieważ w przypadku odnalezienia początkowego obiektu, obiekty z nim powiązane są wyszukane przez IO.

Koncepcja IO jest wszechobecna w obiektowych bazach danych. Tak, jak odpowiadająca jej notacja bieżącego wskaźnika w bazach sieciowych, wspomaga proces przetwarzania danych. Ze względu na ten trwały charakter koncepcji IO mówi się często, że obiektowe bazy danych stanowią kontynuację technologii rozwijanej dla baz sieciowych. W tym kontekście można zgodzić się z taką oceną.

Hermetyzacja

Metody hermetyzują dane i inne metody. Tylko metody umieszczone w publicznym interfejsie obiektu są dostępne dla innych obiektów. W niektórych systemach, w większości przypadków opartych na C++, dane (zmienne, atrybuty) mogą być także umiejscowione w publicznym interfejsie. Taka praktyka nie zawsze jest efektywna. Hermetyzacja ukrywa bowiem szczegóły implementacyjne obiektu, zwiększając w ten sposób niezależność danych (sekcja 2.7) i ułatwiając przyszłe utrzymanie i skalowalność systemu. Hermetyzacja jest także najbardziej obiecującym podejściem w walce z ciągle wzrastającymi trudnościami w dokończeniu aplikacji budowanej w oparciu o odziedziczony system. Poprzez ideę znaną jako „opakowanie obiektu" kompletny COBOL-owy lub inny odziedziczony system może zostać hermetyzowany. Można dla niego zdefiniować interfejs publiczny. W przypadku tworzenia nowych aplikacji mogą one wtedy wymieniać informacje z tak opakowanymi obiektami.

Niestety, siła hermetyzacji jest kompromisem pomiędzy dziedziczeniem i interfejsem zapytań. Należy również odnotować, że hermetyzacja tak, jak jest obecnie implementowana w bazach obiektowych i ich otoczeniu programowym, stosowana jest do obiektów różnych klas. Obiekty z jednej i tej samej klasy nie mogą przecież niczego „ukryć" przed sobą. Słabość hermetyzacji może mieć poważny wpływ na bezpieczeństwo i integralność typowych aplikacji biznesowych.

Generalizacja i dziedziczenie

Technologia obiektowa, a w szczególności obiektowe języki programowania są ukierunkowane na generalizację i dziedziczenie. Obiekty są kategoryzowane na klasy i podklasy, odpowiednio do ich typów. Klasy generalizują swoje podklasy. Podklasa dziedziczy definicję atrybutów i operacji ze swojej klasy rodziców (superklasy). Podklasa może dostarczać różne implementacje dla dziedziczonych operacji. To buduje podstawy dla własności zwanej wielopostaciowością (polimorfizm). Pozwala to na zwiększenie uniwersalności oprogramowania, ponieważ rozszerzenie systemu o nowy obiekt wymaga jedynie zaprojektowania właściwego interfejsu.

Generalizacja stanowi potężną notację semantyczną. Ogranicza redundancję, określa ważne więzy integralności, wspomaga modularność i wielokrotną używalność, ułatwia rozwój systemu i jego utrzymanie. Cała idea tworzenia oprogramowania przez składanie jest oparta na generalizacji i dziedziczeniu. Klasy biblioteczne i gotowe struktury są dostępne dla programistów, a nowe programy są pisane przez dziedziczenie i specjalizację z istniejących już klas, przez ich włączenie do istniejących rozwiązań.

W wielu przypadkach obiektowe bazy danych nie są niczym więcej niż dużymi klasami bibliotecznymi implementującymi funkcje zarządzania bazą i pozwalającymi rozwijać aplikacje przez dziedziczenie i specjalizację z niektórych z tych klas.

Klasy zarządzania bazą danych dostarczają metod do obsługi połączeń bazy danych, operacji zgłoszenia i wyłączenia, sesji bazy danych, długich i krótkich transakcji, wersji i konfiguracji. Normalnie klasy te są względnie zamknięte dla specjalizacji i dziedziczenia przez użytkowników. W ogólności jednak architektura i funkcjonalność obiektowego systemu zarządzania obiektową bazą danych może być dopasowana przez otwarcie tych klas dla użytkowników.

Klasa zarządzania klasami zawiera metody do przetwarzania na całym zbiorze obiektów danej klasy. Taka metoda odpowiadająca relacyjnej instrukcji SELECT

pozwala w odpowiedzi na zapytania wyszukiwać w klasie obiektów IO te obiekty, które spełniają określone kryteria. Inne metody pozwalają: opuścić klasę, opuścić wszystkie obiekty danej klasy, stworzyć indeks, usunąć indeks i in.
Generalizacja i dziedziczenie są realizowane w zmieniającej się architekturze systemu zarządzania obiektową bazą danych. W rezultacie systemy te są bardziej otwarte – zachęcają one programistów piszących aplikacje do rozszerzania systemu przez dziedziczenie i specjalizację. Tych cech brakuje w systemach opartych na modelu relacyjnym. Oczywiście, tak długo jak rozważana jest konstrukcja aplikacji, możliwe jest zdefiniowanie generalizacji jako tablicy w schemacie relacyjnym i wymuszenie dziedziczenia pomiędzy superklasami i podklasami poprzez rozwiązania proceduralne. Jakkolwiek nienaturalne, takie rozwiązanie może być satysfakcjonujące w niektórych zastosowaniach.

Agregacje i wspomaganie multimedialne

Podczas gdy generalizacja jest asocjacją na typach obiektów (klasach), agregacja jest asocjacją na zbiorach obiektów. Agregacja jest naturalną drogą do modelowania aplikacji multimedialnych (tekst, rysunki, głos, wideo i in.). Aplikacje takie narzucają wyraźną strukturę i semantykę przeznaczoną do kompozycji obiektów. Dla przykładu, dokument zawiera rozdziały, rozdziały zawierają sekcje, sekcje zawierają paragrafy, itd. Prowadzi to do zjawiska propagacji w operacjach (np. usunięcie rozdziału pociąga za sobą usunięcie wszystkich sekcji tego rozdziału, itd.). Każdy podzbiór obiektów jest pełnoprawnym obiektem, z własnymi IO, z możliwością niezależnego dostępu przez zapytania lub manipulowania na danych. Obiekty mogą być wyszukiwane przez wartości swoich składowych dzięki użyciu dynamicznie formułowanych zapytań, takich jak np. – „Znajdź wszystkie paragrafy zawierające dane słowo?".

Niestety, praktyka obiektowych baz danych nie włączyła się jeszcze wystarczająco mocno w stosunku do oczekiwań do wdrożenia koncepcji agregacji. Technologia obiektowa koncentruje się na generalizacji i nieco zaniedbuje agregacje. Można mieć nadzieję, że ten trend zostanie w przyszłości odwrócony.

Długie transakcje

Mamy do czynienia z zasadniczą różnicą w sposobie realizacji wielodostępu do danych w architekturze baz relacyjnych i obiektowych. Bazy relacyjne są projektowane do wspomagania wielodostępu. Bazy obiektowe są natomiast projektowane tak, aby wspomagały wielodostęp. Zamiast więc izolować użytkowników, są oni integrowani. Powinni mieć świadomość tego, że współdzielą dane. Krótkie transakcje są nadal dostępne, lecz są elementami długich transakcji, które mogą trwać tak długo, jak grupy robocze pracują razem nad rozwiązaniem zadania (np. tak długo, jak architekci projektują most).
Relacyjne bazy danych nie wspomagają długich transakcji i dlatego nie są wygodne przy opracowywaniu aplikacji dla pracy grupowej. Bazy obiektowe wspomagają zarówno krótkie, jak i długie transakcje, i dlatego (przynajmniej potencjalnie) mogą być używane dla projektowania transakcji *on line* występujących w typowych aplikacjach biznesowych, takich jak np. systemy finansowo-księgowe.

Wersje i konfiguracje

Długie transakcje w obiektowych bazach danych są wspomagane przez wersje i konfiguracje. Obiekt może mieć wersje. Wersja obiektu jest zawsze dostępna przy jego ściąganiu z grupowej bazy danych. Nowa wersja jest tworzona podczas operacji wyprowadzania obiektu. Co więcej, dzięki różnym wersjom można śledzić ewolucję stanów obiektu.
Idea wersji i konfiguracji współgra z koncepcją agregacji i wspomagania multimedialnego. Dla przykładu nowa wersja superzbioru obiektów może być stworzona zawsze, kiedy nowa wersja podzbioru obiektów jest dodawana do bazy danych. Ta koncepcja jest czasami nazywana filtrowaniem (percolation).

Sprawność

Długa droga rozwoju technologii relacyjnych baz danych od koncepcji sformułowanej w 1970 roku do akceptacji przez użytkowników w połowie lat osiemdziesiątych spowodowana była ich niewystarczającą wydajnością. W tym świetle twierdzenie o wielkiej sprawności baz obiektowych może wydawać się trochę podejrzane. Jest również prawdą, że wiele porównawczych testów wydajności, mówiących o kilkuset razy większej sprawności baz obiektowych, przedstawia obraz trochę jednostronny i ma na celu pokazanie siły technologii obiektowej. Przewaga sprawności baz obiektowych jest jednak faktem.
Obiektowe bazy danych mogą znacznie przewyższyć sprawność systemów z bazami relacyjnymi w aplikacjach z częstą i cykliczną nawigacją pomiędzy połączonymi obiektami załadowanymi do pamięci. Oczekuje się tego od większości aplikacji obsługi baz danych, a szczególnie konwencjonalnych aplikacji dla biznesu. Oczywiście sprawne są tylko systemy dobrze zaprojektowane i skalowalne.

Rywalizacja czy koegzystencja?

Nie ma prostej odpowiedzi na pytanie postawione w tym artykule. Z jednej strony obiektowe bazy danych są zagrożeniem dla relacyjnych, ponieważ mają szereg pożądanych cech, których nie mają systemy baz relacyjnych. Z drugiej strony, relacyjne bazy danych są zagrożeniem dla obiektowych, ponieważ stoi za nimi duża siła finansowa ich producentów oraz dlatego, że wiele cech obiektowych może być implementowanych lub symulowanych w systemach relacyjnych.

dokończenie na s. 26 →

Jak płacić programistom?

Grażyna Hołodnik-Janczura

Problem oceny potrzebnych nakładów i przewidywania terminów realizacji systemu informatycznego jest nieodłącznym zagadnieniem planowania i kierowania przedsięwzięciem tworzenia oprogramowania. Często znacznie przekraczane są planowane na początku koszty, terminy nie są dotrzymywane oraz efekty nie odpowiadają założeniom. Zmniejszenie ryzyka takich przedsięwzięć można osiągnąć, podejmując decyzje opierając się na wymiernych wskaźnikach. Metody szacowania kosztów umożliwiają ustalenie potrzebnej wielkości nakładów w przeliczeniu na liczbę odpowiednich jednostek czasu (np. roboczogodzin lub osobomiesięcy miesięcy) dla określonej wielkości planowanego systemu.

Najczęściej stosowanymi metodami wyceny pracy programisty są FPA I COCOMO.

Warunkiem przewidywania potrzebnych nakładów do zrealizowania podejmowanego przedsięwzięcia informatycznego jest określenie **zakresu planowanego systemu w aspekcie realizowanych funkcji i przetwarzanych informacji.** Okazuje się, że miara tego zakresu jest najważniejszym problemem istniejących metod szacowania kosztów takiego przedsięwzięcia.

Aktualnie stosowane są metody będące modyfikacjami modeli pochodzących z przełomu lat 70-tych i 80-tych. Biorąc pod uwagę założenia tych metod i sposoby ich powstawania można przeprowadzić następującą klasyfikację ocenianych wielkości. Ze względu na obecny we wszystkich metodach i podstawowy ich element – rozmiar tworzonego oprogramowania, metody szacowania kosztów można podzielić na dwie grupy:

- pierwsza, gdzie zarówno wielkość tworzonego programu, jak i wskaźnik produktywności (liczba jednostek wielkości programu przypadająca na jednostkę czasu) są określane za pomocą **liczby linii kodu źródłowego programu** (LOC – *lines of cod*) albo też **liczby instrukcji źródłowych** (DSI – *delivered source instructions*);
- druga grupa metod to modele kosztów, w których miarą wielkości tworzonego programu (systemu) są **punkty funkcyjne**, określające jego rozmiar przez identyfikację składowych systemu, postrzeganych w aspekcie realizowanych funkcji użytkownika (np. liczba wejść, liczba plików).

Jeżeli wielkość oprogramowania jest określana liczbą linii kodu źródłowego programu lub liczbą instrukcji, to:

- szacowanie potrzebnych nakładów, które odbywa się na podstawie samego kodowania programu (a jest to tylko jeden z etapów realizacji [1]) wytwarza wskaźnik w znacznym stopniu uproszczony i z dużym przybliżeniem;
- możliwość wykonania takiego szacowania jest możliwa dopiero wtedy, gdy znana jest wielkość programu, a zatem jest to dużo za późno, jeżeli chcielibyśmy na tej podstawie określić wielkość planowanych nakładów;
- miara w postaci liczby linii kodu utrudnia wykonywanie badań porównawczych. Zarówno w przypadku liczby instrukcji, jak i liczby linii kodu ich identyfikacja nie zawsze jest jednoznacznie określona. Jest w pełni zależna od środków implementacji, ponieważ ten sam problem, rozwiązywany za pomocą programu napisanego w różnych językach programowania, będzie przedstawiał inną wielkość.

Jeżeli metoda pozwala ustalić wielkość oprogramowania za pomocą punktów funkcyjnych, to:

- jest niezależna od środków implementacji,
- jest możliwa do zastosowania na wcześniejszych etapach cyklu życia, kiedy to na podstawie analizy wymagań użytkownika przyszłego systemu można określić jego przybliżoną wielkość.

Tabela 1. Klasyfikacja metod szacowania kosztów wg miar i typów modeli.

Model	Miara	Typ modelu
FPA	punkty funkcyjne	empiryczny, statyczny
MARK II	punkty funkcyjne	empiryczny, statyczny
COCOMO	liczba instrukcji	hybrydowy
SLIM	liczba linii kodu	dynamiczny
Pfleegera	liczba linii kodu	statyczny
Halsteada	liczba linii kodu	teoretyczny,statyczny

Metoda FPA

Metoda „Analizy punktów funkcyjnych" (*Function Point Analysis* – FPA) została opracowana przez Allana Albrechta dla firmy IBM i opublikowana w 1979 roku. Jest to metoda empiryczna, otrzymana w wyniku doświadczeń przeprowadzonych na 24 aplikacjach. W celu obliczenia kosztu tworzenia oprogramowania ustalono kolejne etapy obliczeń, prowadzących do otrzymania wskaźnika produktywności, wyrażającego stosunek wielkości produktu do poniesionego nakładu pracy. Nakład pracy mierzono roboczogodzinami, natomiast wielkość produktu, czyli systemu informatycznego, określono za pomocą punktów funkcyjnych. Punkty funkcyjne są wielkością bezwymiarową, wyrażającą istotne składniki systemu, poziom ich złożoności oraz stopień wpływu wybranych charakterystyk systemu na jego wielkość.

Wybór punktów funkcyjnych jest niezależny od narzędzi stosowanych w tworzeniu oprogramowania, jak i od narzędzi implementacyjnych, ale zastosowanie metody „Analizy punktów funkcyjnych" jest uzależnione od dziedziny, z jakiej pochodzi rozwiązywany problem. Zgodnie z założeniami autora metoda ta była przeznaczona do szacowania kosztów tworzenia systemów informatycznych, służących do zarządzania firmą. Stosowana do

projektów tworzenia aplikacji naukowych i technicznych, a więc zawierających skomplikowane algorytmy obliczeniowe, nie daje prawidłowych wyników.
W celu ustalenia punktów funkcyjnych są brane pod uwagę dwa podstawowe elementy, a więc:
- wielkość przetwarzanych informacji,
- poziom technicznej złożoności.

Metoda FPA ma trzy etapy:

I. Obliczanie niedopasowanych punktów funkcyjnych UFP (Unadjusted Function Points).

W pierwszym kroku jest ustalany rozmiar systemu przez identyfikowanie jego części składowych, widzianych przez końcowego użytkownika. Następnie przeprowadza się klasyfikację tych elementów według pięciu określonych typów. Znaczenie składników systemu dla mierzonej wielkości produktu jest wyrażane za pomocą liczby wagowej. Wielkości wagowe danego typu składników zależą od poziomu ich złożoności (prosty, średni i złożony). Kryterium tego podziału uwzględnia przede wszystkim liczbę elementów danych w każdym typie składnika. Zatem każdy składnik jest przedstawiany jako liczba punktów zależna od jego typu i poziomu złożoności. Natomiast suma punktów dla wszystkich składników, która stanowi tzw. „niedopasowane" punkty funkcyjne UFP.

II. Obliczanie sumarycznego poziomu technicznej złożoności TCF (The Technical Complexity Factor).

Czternaście czynników, przedstawionych w tab.2, zwanych „głównymi charakterystykami aplikacji", stanowi grupę elementów o drugorzędnym znaczeniu dla szacowanego rozmiaru systemu, ale ma istotne znaczenie dla oceny kosztów tworzenia oprogramowania. Odzwierciedlają one parametry techniczne związane z realizacją przetwarzania danych.

III. Obliczanie punktów funkcyjnych FP (Function Points).

Po wykonaniu rachunków w obu etapach, wielkość systemu jest ustalana w punktach funkcyjnych.
Przeprowadzone obliczenia[2] w ramach tych trzech etapów pozwalają na:
- odczytanie dla wyliczonej liczby punktów funkcyjnych czasu potrzebnego na realizację takiego systemu (z tabeli dostarczanej wraz z metodą lub opracowanej na podstawie własnych doświadczeń), a następnie
- obliczenie wskaźnika produktywności jako stosunku liczby punktów funkcyjnych do czasu realizacji systemu.

Otrzymany wskaźnik produktywności stanowi podstawę do sterowania i kontroli realizacji oprogramowania.

Metoda MARK II Function Points

Rezultatem alternatywnego podejścia do zastosowania metody punktów funkcyjnych Albrechta jest metoda MARK II, opracowana przez Ch. Symonsa i promowana jako nowa miara wielkości przetwarzanych informacji. Zachowuje ona podstawowe przesłanki metody Albrechta, tj. niezależność wielkości przetwarzanych informacji od technologii przetwarzania (w miarę możliwości) oraz zachowanie terminologii łatwo rozpoznawalnej przez użytkownika systemu.

Podstawowe założenia metody MARK II:
- system jest postrzegany jako zestaw logicznych typów transakcji, takich jak logiczne kombinacje wejście/proces/wyjście,
- pliki przekazywane lub wspólne z innymi systemami są traktowane jako innego rodzaju wejście lub wyjście,
- zapytania są rozważane jako innego rodzaju kombinacja wejścia/procesu/wyjścia,
- koncepcja „logicznego pliku wewnętrznego" z metody Albrechta jest zastąpiona pojęciem „entity" (w literaturze polskiej znane jako encja lub obiekt[3]).

Założenia te modyfikują podejście do klasyfikacji składowych wielkości systemu, wprowadzając pojęcie obiektu jako nowy element systemu. Zaproponowane składniki i skala wielkości systemu są związane z nakładem pracy na etapie analizy, projektowania i tworzenia funkcji systemu. Takie podejście jest zrozumiałe i adekwatne z poziomu pracy twórców systemu, a nie jak w metodzie Albrechta, uzależnione od przekazów użytkownika systemu.

Metoda COCOMO

Strukturalny model kosztów COCOMO (*COnstruktive COst MOdel*) jest modelem hybrydowym, ponieważ łączy metody doświadczalne z elementami wnioskowania statystycznego. Jest metodą trójpoziomową: na poziomie

Tabela 2. Lista charakterystycznych czynników aplikacji wg metody FPA.

ID	CHARAKTERYSTYKA	ID	CHARAKTERYSTYKA
C1	Transmisja danych	C8	Aktualizacja on-line
C2	Rozproszone przetwarzanie	C9	Stopień złożoności przetwarzania
C3	Wydajność	C10	Wielokrotność zastosowania
C4	Obciążenie konfiguracji	C11	Łatwość instalacji
C5	Wskaźnik ilościowy transakcji	C12	Łatwość eksploatacji
C6	Wprowadzanie danych on-line	C13	Wielokrotność stanowisk
C7	Umiejętności końcowego użytkownika	C14	Możliwość wprowadzania zmian

podstawowym, pośrednim i szczegółowym. Modele budowane na kolejnych poziomach tworzą hierarchię coraz to bardziej szczegółowych modeli, od modelu w skali makro-estymacji, jako funkcji wielkości produktu, do modelu mikro-estymacji. Ponadto metoda COCOMO zawiera formuły dla przewidywania czasu tworzenia projektu, dzieląc nakłady na fazy i czynności. Pojęcie projektu jest tutaj używane w znaczeniu przedsięwzięcia informatycznego.

Metoda COCOMO poziomu pośredniego przebiega następująco:

1. Nominalny nakład tworzenia jest estymowany jako funkcja wielkości produktu, mierzonej w tysiącach instrukcji kodu źródłowego *KDSI (Kilo Delivered Source Instructions)* oraz ustalonego trybu tworzenia projektu.
2. Zestaw współczynników nakładów określa się na podstawie wskaźników produktu ze zbioru piętnastu atrybutów sterujących kosztem.
3. Estymowany nakład tworzenia otrzymuje się przez pomnożenie nominalnej wielkości nakładu przez wszystkie współczynniki.
4. W celu obliczenia, na podstawie przeprowadzonej estymacji, wielkości nakładu w jednostkach monetarnych należy uwzględnić także dodatkowe elementy, np. harmonogram tworzenia, fazę i czynności dystrybucji, koszty komputera, koszty pielęgnacji.

Krok 1. Estymacja nominalnego nakładu.

Na podstawie cech, wymienionych w tabeli 3, uzgadnia się profil tworzonego projektu.
Po przeanalizowaniu 63 projektów autor metody otrzymał dla każdego z trybów różne parametry równania estymacji.

Krok 2. Określenie współczynników nakładu.

Sformułowano piętnaście czynników uznanych za atrybuty sterujące kosztem. Po ustaleniu sześciu kategorii poziomu ich wpływu (bardzo niski, niski itd.) oraz opracowaniu odpowiedniej skali otrzymano doświadczalnie zbiór odpowiadających im współczynników:

Dla znalezienia sumarycznego czynnika wnioskowania w pierwszej kolejności należy ustalić poziom wpływu (niski, wysoki itd.) każdego z atrybutów. Można obliczyć iloczyn wszystkich współczynników, który wskazuje, ile razy musi być pomnożona wielkość nominalnego nakładu danego projektu w celu odzwierciedlenia udziału w kosztach czynników innych niż tylko rozmiar oprogramowania.

Tabela 4. Sterowniki kosztu w metodzie COCOMO.

Atrybuty produktu		
	RELY	Wymagana niezawodność oprogramowania
	DATA	Wielkość bazy danych
	CPLX	Stopień złożoności produktu
Atrybuty komputera		
	TIME	Limit czasu wykonania
	STOR	Limit pamięci
	VIRT	Lotność maszyny wirtualnej
	TURN	Czas cyklu komputera
Atrybuty personelu		
	ACAP	Kwalifikacje analityków
	AEXP	Doświadczenia aplikacyjne
	PCAP	Kwalifikacje programistów
	VEXP	Doświadczenie z maszynami wirtualnymi
	LEXP	Doświadczenie w programowaniu w danym języku
Atrybuty projektu		
	MODP	Praktyka w stosowaniu nowoczesnego programowania
	TOOL	Stosowanie oprogramowania narzędziowego
	SCED	Wymagany harmonogram tworzenia

Krok 3. Estymacja nakładu tworzenia.

Przybliżony nakład tworzenia oblicza się jako iloczyn

Tabela 3. Tryby tworzenia oprogramowania w metodzie COCOMO.

CECHA	TRYB		
	ORGANICZNY	BLIŹNIACZY	WBUDOWANY
Organizacyjne zrozumienie celów produktu	Kompletne	Znaczne	Ogólne
Doświadczenie w pracy z podobnymi systemami	Obszerne	Znaczne	Umiarkowane
Potrzeba dostosowania oprogramowania do wcześniej ustalonych wymagań	Podstawowa	Znaczna	Pełna
Potrzeba dostosowania oprogramowania do specyfikacji zewnętrznego interfejsu	Podstawowa	Znaczna	Pełna
Równoległość rozwoju oprogramowania w powiązaniu z nowym sprzętem i operacyjnymi procedurami	Niektóre	Umiarkowanie	Obszernie
Potrzeba innowacji struktury danych i algorytmów ich przetwarzania	Minimalna	Niektóre	Znaczna
Premia za wcześniejsze zakończenie	Niska	Średnia	Wysoka
Zakres wielkości produktu	<50 KDSI	<300 KDSI	Dowolna wielkość

nominalnego nakładu tworzenia i sumarycznego poziomu wpływu piętnastu sterujących atrybutów kosztu.

Metoda COCOMO umożliwia pokazanie dodatkowych zależności związanych z tworzeniem oprogramowania w celu podania oszacowanych nakładów projektu w jednostkach monetarnych, a także dokonania rozkładu kosztów i nakładów na fazy cyklu życia (analiza wymagań, projektowanie itd.) lub podziału ze względu na występujące w projektach rodzaje czynności (programowanie, planowanie testu, zarządzanie itd.). Te uzupełniające relacje wspomagają estymację budowanego harmonogramu projektu oraz fazę jego dystrybucji.

Model Pfleegera

Model opracowany w 1988 roku przez S.L. Pfleegera ujmuje, bardziej szczegółowo niż inne modele, zagadnienie tzw. ponownego wykorzystania w innym projekcie tych samych modułów programu, czy też bibliotek oprogramowania. Jest to model statyczny, który wyraża rzeczywistą produktywność projektu jako wynik iloczynu jego średniej produktywności i funkcji regulującej, zależnej od całego zestawu czynników kosztu.

W odróżnieniu od innych modeli, zamiast wyboru ze stałego zestawu czynników autor proponuje zdefiniowanie własnych czynników, odpowiadających specyfice projektu i środowiska, w których jest realizowany. Dla każdego zidentyfikowanego czynnika X należy określić współczynniki poprzez badanie kosztu włączenia X do projektu i ustalenia liczby projektów, powyżej której koszty te będą zamortyzowane. Np. jeśli X jest wykorzystywanym narzędziem, to współczynnik A_X dla tego narzędzia jest wyprowadzany na podstawie:

- części projektu stosującej narzędzie X (od 0 do 1),
- kosztu tworzenia narzędzia X,
- kosztu włączenia narzędzia X do projektu, wliczając koszt szkolenia twórców w jego stosowaniu,
- liczby projektów, powyżej której koszty będą zamortyzowane, tj. przewidywana liczba projektów które będą wykorzystywały narzędzie X.

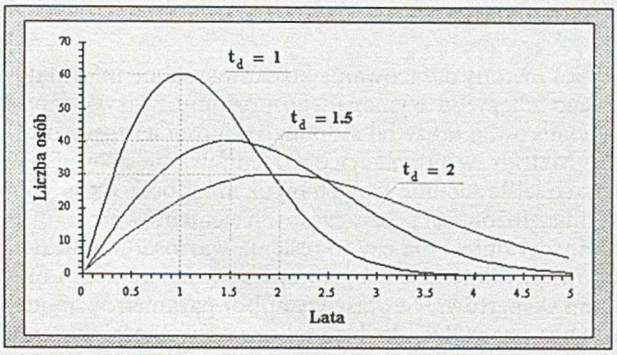
Oszczędności mogą powstać w przypadku wielokrotnego wykorzystywania wytworzonego modułu programowego. Wielkość nakładów będzie inna, gdy badany moduł będzie zastosowany w wielu projektach, niż gdy będzie to moduł uniwersalny, włączony do biblioteki programów. Model Pfleggera umożliwia przeprowadzanie analiz porównawczych w celu określenia liczby wymaganych zastosowań zapewniających opłacalność wykonania wszystkich projektów. Rozważania te proponuje się prowadzić nie tylko w aspekcie możliwości wielokrotnego stosowania tego samego kodu źródłowego programu, ale także innych czynników wpływających na koszt produkcji oprogramowania, np.ujednolicenie specyfikacji wymagań, projektu bazy danych czy testowania danych.

Metoda SLIM

Model metody SLIM (Software Life cycle Management) powstał w 1977. Opracował go L. Putnam na podstawie analizy ponad 50 projektów wojskowego ośrodka informatycznego oraz 150 innych. Najważniejszą przesłanką tej metody jest ciągła zmienność tempa tworzenia kodu programu. Za istotne dla nakładu tworzenia uznano następujące czynniki:

- czynnik technologii projektu mierzony doświadczeniem zespołu,
- poniesiony nakład pracy,
- ustalony czas tworzenia,
- dowolny punkt czasowy w cyklu życia (zmienna niezależna),
- gradient trudności mierzony złożonością problemu.

Dynamiczna formuła tego modelu umożliwia badanie wpływu liczby osób, zaangażowanych w realizację projektu, na spodziewany czas jego zakończenia.



Wykres. Alternatywne nakłady na realizację projektu

Krzywe, przedstawione na wykresie, ilustrują trzy możliwe rozkłady nakładu pracy w czasie dla tego samego projektu. Całkowita wielkość nakładu, reprezentowana przez obszar pod każdą krzywą jest taka sama, ale wielkość nakładu, po upływie tego samego czasu, ma dla każdej z nich inną wartość.

Model teoretyczny Halsteada

W roku 1977 w opracowaniu Halsteada sformułowano kilka zasad poznawania i mierzenia charakterystyk oprogramowania dotyczących obliczania operatorów i operandów występujących w module programowym. Zaproponowano pomiar złożoności projektu zakładając, że bardziej złożone projekty są bardziej kosztowne. Przyjęto dwa główne założenia o tworzeniu systemu:

- po pierwsze, wyrażenie algorytmu, który ma być implementowany w rozważanym module, jest kompletne i jednoznaczne;
- po drugie, programista, piszący kod, zna biegle język implementacji, będzie pracował samodzielnie i potrafi się skoncentrować na pracy bez dłuższych przerw.

Jako wskaźnika wielkości wymaganego nakładu użyto pojęcia liczby „pamięciowych wyróżnień", którą oblicza się korzystając z liczby operatorów i operandów występujących w obliczeniach.

Uwagi praktyczne

Rola metod szacowania kosztów w zarządzaniu przedsięwzięciem informatycznym

W wyniku przeprowadzenia działań, zgodnych z wybraną metodą, otrzymuje się albo wielkość nakładów pracy potrzebnych do zrealizowania danego oprogramowania w jednostkach czasowych albo wskaźnik produktywności dla zespołu realizującego projekt. Dane te mogą stanowić

podstawę planowania przedsięwzięcia informatycznego, a następnie oszacowania pracochłonności zadań i budowy szczegółowego harmonogramu przydzielania tych zadań wykonawcom. Harmonogram może być wygenerowany za pomocą dowolnego programu do zarządzania przedsięwzięciem. W połączeniu z metodami estymacji będzie można na bieżąco śledzić przebieg realizacji projektu i ewentualnie modyfikować wielkość oszacowanej pracochłonności, korygować harmonogram i wprowadzać zmiany w zaplanowanym budżecie.

Wnioski i zalecenia

Wybór metody oszacowania kosztów systemu informatycznego zależy od wymaganego poziomu wiarygodności jej wyników, a także od stopnia trudności jej zastosowania. Metody wspomagające podejmowanie decyzji, nie mające ściśle zdefiniowanych reguł, np. sposób określenia stopnia złożoności poszczególnych elementów tworzonego oprogramowania czy określenia wartości współczynników, powinny być uzupełnione doświadczeniem i opiniami ekspertów. Niewłaściwy dobór parametrów modelu może prowadzić do błędnej oceny wielkości systemu i do estymacji kosztów znacznie odbiegającej od rzeczywistych.
W literaturze najczęściej omawiane są zastosowania metod, w których jednostką pomiaru wielkości systemu jest punkt funkcyjny oraz trójpoziomowa metoda COCOMO. Należy podkreślić, że metody punktów funkcyjnych mogą być stosowane tylko do informatycznych systemów zarządzania. Metoda COCOMO jest natomiast bardziej uniwersalna i może być stosowana do systemów technicznych i naukowych.

Zarówno metoda FPA, jak i COCOMO powstały w wyniku przeprowadzenia wielu eksperymentów. Uwzględniono także przy ich opracowaniu opinie uczestniczących w nich kierowników projektów. Dla tych metod ustalono arbitralnie charakterystyki aplikacji (metoda FPA), atrybuty sterujące kosztem (metoda COCOMO), a ich współczynniki oraz parametry występujące w równaniach zostały znalezione doświadczalnie. Wymienione zaś elementy są charakterystyczne dla organizacji biorących udział w testowaniu lub podobnych. Zatem w przypadku zastosowania ich do innego zespołu projektującego powinny być zweryfikowane.

* * *

Oczekiwania zarówno użytkowników tych metod, jak i ich twórców, idą w kierunku poszukiwania metod niezależnych od subiektywnych decyzji oraz możliwych do zastosowania na wczesnych etapach tworzenia systemu. Aktualnie tworzone systemy informatyczne, zgodnie z nowoczesną technologią tworzenia, podsuwają możliwości obiektywnego podejścia do ustalania wielkości tworzonego systemu i wykonania estymacji wcześniej, niż jest to możliwe przy tradycyjnym podejściu. Podstawą tych technologii jest gromadzenie informacji na temat wymagań użytkownika w postaci sformalizowanych struktur już od samego początku prowadzenia prac. Informacje te, czyli wiedza o powstającym systemie, są zapamiętywane w tzw. repozytorium, czyli metabazie systemu informacyjnego. Metabaza jest skarbnicą danych, uszczegółowianą w miarę postępu prowadzonych prac. Wykorzystanie tej skarbnicy wiedzy dla potrzeb metod szacowania kosztów pozwoli, już na wczesnych etapach realizacji przedsięwzięcia informatycznego, ustalić przybliżoną wielkość planowanego systemu oraz uniezależnić metodę od subiektywnych opinii eksperta. Zatem może okazać się realne przeprowadzanie wczesnej i obiektywnej estymacji nakładów bez dodatkowego nakładu pracy związanego z gromadzeniem informacji oraz bez konieczności ich dodatkowej standaryzacji. Jednocześnie jest to szansa na znaczne zwiększenie automatyzacji omawianego procesu estymacji.

Z istniejących metod, jedynie te, w których obraz systemu modeluje się za pomocą punktów funkcyjnych, nadają się do zastosowania nowoczesnego podejścia. Dokonano już kilka prób wykorzystania możliwości dostarczanych przez narzędzia inżynierii systemów CASE (*Computer Aided System Engineering*). Jednak takie rozwiązania wiążą się z koniecznością poniesienia dużych nakładów finansowych, najpierw na zakup narzędzi inżynierii systemów, a następnie narzędzi do estymacji nakładów potrzebnych na realizację planowanego przedsięwzięcia informatycznego.

---

[1] Zgodnie z metodyką tworzenia systemów informatycznych CASE*Method koszty etapu budowy systemu (kodowanie i tworzenie bazy danych) stanowią 35% kosztów całego przedsięwzięcia).

[2] Przykład obliczania punktów funkcyjnych i produktywności zgodnie z metodą „Analizy punktów funkcyjnych" można znaleźć w opracowaniu.

[3] Obiekty funkcjonują od wielu lat w procesie modelowania systemów informacyjnych jako podstawowy element diagramu Obiekt – Związek (Entity Relationship Diagram), stanowiącego model konceptualny danych, który odzwierciedla potrzeby informacyjne użytkownika i jest budowany od samego początku prac związanych z tworzeniem systemu. Najpierw jako model ogólny, a na etapie szczegółowej analizy wymagań użytkownika, jako model szczegółowy.

Relacyjne a obiektowe...

dokończenie ze s. 21

Możliwość koegzystencji relacyjnych i obiektowych baz danych dowodzi istnienie baz hybrydowych np.: UniSQL, Illustra, MATISEE, HP Odapter. Niektóre z tych systemów są wyposażone w interfejs obiektowy do innych baz relacyjnych, inne w interfejs SQL do baz obiektowych. Wiele z tych systemów udowodniło swą przydatność dla użytkowników, ponieważ pozwalają one na płynne przejście między różnymi rozwiązaniami.
Podsumowując, nie można dać jednej i jednoznacznej odpowiedzi na pytania postawione w tytule. Możemy powiedzieć, że nikt nie ma gotowych odpowiedzi. Chcieliśmy tylko dostarczyć Czytelnikom argumentów do porównania różnych technologii baz danych i krytycznego spojrzenia na ich rozwój.

Witold Abramowicz i Grzegorz Bartoszewicz
Akademia Ekonomiczna w Poznaniu
Katedra Informatyki Ekonomicznej
Leszek A. Maciaszek
Macquarie University, Sydney
Akademia Ekonomiczna w Poznaniu
Katedra Informatyki Ekonomicznej

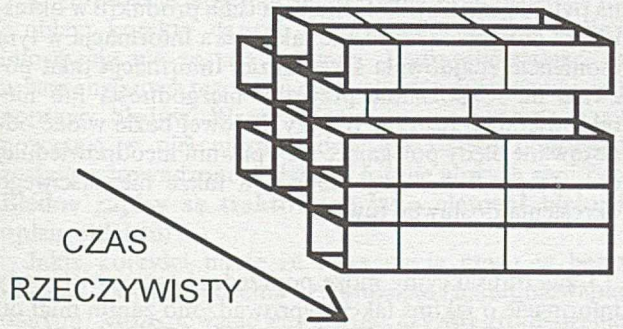
Systemy temporalnych baz danych

Krzysztof Kania
Jerzy Gołuchowski

W poprzednich naszych artykułach [1] i [2] przedstawiliśmy trudności oraz możliwości ujęcia danych zmiennych w czasie na różnych poziomach systemu baz danych. W niniejszym artykule – rozwijając uproszczony model hurtowni – przedstawiono możliwe sposoby rozbudowy tradycyjnej (chwilowej) bazy danych tak, aby mogła służyć do przechowywania danych temporalnych. Na prostych przykładach przedstawiono ideę różnych rodzajów temporalnych baz danych (TBD) i korzyści płynące z zastosowania takich baz. W przykładach posłużono się tradycyjną relacją rozbudowując ją o dodatkowe atrybuty.

Rodzaje temporalnych baz danych

Praktyka wskazuje, że użytkownicy baz danych najczęściej są zainteresowani tym, aby informacje zawarte w bazie danych jak najlepiej odzwierciedlały przeszłe i obecne, a niekiedy i przewidywały przyszłe stany rzeczywistości. Bazy danych, pozwalające odtworzyć przyszłe stany, które miały miejsce w rzeczywistości, określa się jako **historyczne bazy danych** (historical databases) (rys. 1).



Rys. 1. Graficzna reprezentacja relacji historycznej. Źródło: [7]

Historyczne bazy danych (HBD) pozwalają na zapis czasu rzeczywistego tzn. takiego, w którym konkretna wartość danego atrybutu według naszej najlepszej wiedzy (!) występowała w rzeczywistości (np. [7, 9]). Taka jednoznaczna definicja wymiaru czasu pozwala na przeniesienie działań związanych z jego obsługą (definicja bazy, aktualizacja, zapytania) na poziom SZBD. Należy zaakcentować fakt, że bazy takie są nie malejącymi kolekcjami danych, co oznacza ich nieustanny rozwój, tak w sensie logicznym, jak i fizycznym. Podstawowym typem zapytań w przypadku HBD jest pytanie o historię wartości atrybutów obiektów opisanych w bazie.

Jeden z możliwych obrazów takiej bazy dla rozpatrywanego przykładu ujmuje relacja Towar1 przedstawiona w Tabeli 1. Zapisanie informacji w ten sposób umożliwia jednoznaczne opisanie faktów, których do tej pory nie można było do bazy wprowadzić, np.:

Dostawca A zaprzestał dostaw 30.04.94, a inny (B) pracuje od 20.05.94 do chwili obecnej.

Pomiędzy 26.04.94 a 30.04.94 nie była znana cena łyżek (nie powinno być żadnych transakcji sprzedaży).

Brak danych na temat obrotu łyżkami i widelcami w pewnych okresach czasu.

W bazie przedstawionego typu można informację o zdarzeniach, które dopiero wystąpią. W zależności od daty bieżącej ostatnią informację o zdarzeniach, które dopiero wystąpią. W zależności od daty bieżącej ostatnią informację np. o cenie widelców można interpretować jako sezonową zmianę cen, mającą nastąpić w okresie od 05.06.95 do 01.09.94. Jeśli po tej dacie w bazie brak kolejnych zapisów to należy przypuszczać, że zaniedbano określenia nowej ceny tego towaru.

Ze względu na wysoką redundancję, wynikającą z konieczności przepisywania całych krotek nawet po zmianie jednego atrybutu temporalnego, w modelu takim dokonuje się tzw. normalizacja TBD tak, aby jedna dana temporalna znalazła się w jednej relacji. Obraz przekształconej w ten sposób bazy ujmują relacje Cena2 i Dostawca 2 w Tabeli 2.

Tabela 1. Relacja: Towar1

Towar	Cena	Data pocz.	Data końc.	Dostawca	Data pocz.	Data końc.
łyżki	15	01.03.94	25.04.94	A	01.03.94	30.04.94
łyżki	20	01.05.94	31.05.94	B	20.05.94	nadal
łyżki	30	01.06.94	nadal	B	20.05.94	nadal
widelce	30	10.03.94	nadal	A	05.05.94	20.05.94
widelce	30	10.03.94	nadal	B	25.05.94	04.06.94
widelce	25	05.06.94	01.09.94	A	25.06.94	nadal
noże	40	01.05.94	nadal	C	01.05.94	nadal

Tabela 2. Relacja: Cena2

Towar	Cena	Data pocz.	Data końc.
łyżki	15	01.03.94	25.04.94
łyżki	20	01.05.94	31.05.94
łyżki	30	01.06.94	nadal
widelce	30	10.03.94	nadal
widelce	25	05.06.94	01.09.94
noże	40	01.05.94	nadal

Relacja: Dostawca2

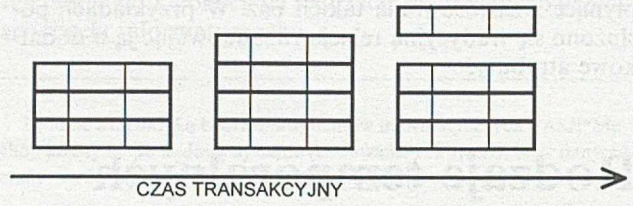
Towar	Dostawca	Data pocz.	Data końc.
łyżki	A	01.03.94	30.04.94
łyżki	B	20.05.94	nadal
widelce	A	05.05.94	20.05.94
widelce	B	25.05.94	04.06.94
widelce	A	25.06.94	nadal
noże	C	01.05.94	nadal

Podział relacji owocuje oszczędnością miejsca, lecz komplikuje sposób uzyskania odpowiedzi na niektóre pytania, szczególnie na te, które wymagają powiązania zmieniających się wartości ceny i nazw dostawców. W rozwiązaniu tym nadal występuje redundancja i dlatego wielu autorów skłania się ku wykorzystaniu modeli, które nie są oparte o postać normalną bazy danych [11]. Ze względu na inny cel tego opracowania problemy implementacyjne TBD nie będą omawiane.

W większości przypadków uwzględnienie zapisu czasu, w którym faktycznie atrybuty przybierały określone wartości, jest wystarczające. Zdarzają się jednak sytuacje, gdy zachodzi bezwzględna potrzeba zapisu informacji o czasie zapisu informacji w systemie, a także o popełnionych błędach, czasie ich powstania i wprowadzenie poprawek. Dysponując bazami historycznymi i stwierdzając, że na przykład cena widelców została w bazie niewłaściwie zapisana, można oczywiście poprawić zapis, lecz poprzedni błędny zapis zostaje skasowany.

Brak możliwości pozostawienia błędnych danych w bazie systemu może prowadzić do tego, że aby dociec źródeł nietrafnych decyzji sięga się do tradycyjnych kartotek i archiwów. Zdaniem wielu osób (np. rewidentów SI rachunkowości) ogranicza to praktyczną użyteczność systemów komputerowych. System, w którym nie znajdują się zapisy o popełnionych błędach, nie spełnia też w pełni założeń systemu wspomagającego decyzje, ponieważ często nie jest w stanie wskazać podjętych wcześniej, a zaniechanych działań.

W pewnych przypadkach lub dla niektórych użytkowników (np. administratora systemu, rewidenta) interesujący może być jedynie czas, w którym odpowiednie zapisy znalazły się w bazie, niezależnie od tego, kiedy fakt miał miejsce w rzeczywistości. Bazy pozwalające odtworzyć historię stanów systemu określane są jako **transakcyjne bazy danych** (transaction databases, rollback databases) (rys. 2), zaś czas, w którym nastąpiła zmiana bazy jako czas transakcyjny (transaction time). Kolejne wersje bazy reprezentują sekwencję chwilowych jej ujęć, odzwierciedlających w całości działanie bazy danych od momentu jej powstania do chwili obecnej. Tak więc podstawowym typem pytań jest pytanie o przeszły stan bazy.



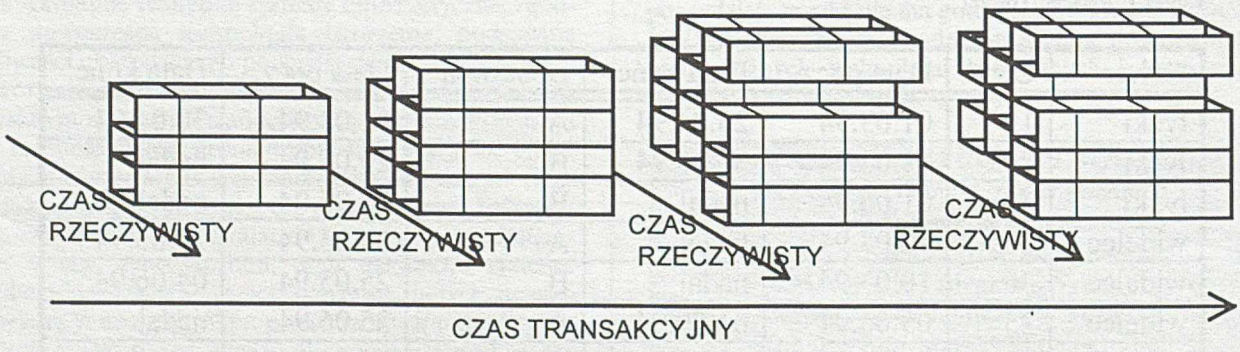
Rys. 2. Graficzna reprezentacja relacji transakcyjnej. Źródło. [7]

Bazy danych, które umożliwiają zapis zarówno czasu rzeczywistego, jak i transakcyjnego, są określane jako **temporalne** (temporal databases) lub dla podkreślenia – **bitemporalne bazy danych** (bitemporal databases) (rys. 3).

W rozpatrywanym przykładzie relacje spełniające warunki modelu bitemporalnej bazy danych przyjmują postać przedstawioną w tabeli 3.

Takie zapisy w bazie pozwalają nie tylko odpowiedzieć na pytanie, na przykład jaka była cena produktu w określonym momencie czasu, ale także jaka informacja w tym momencie znajdowała się w bazie. Informacja taka pozwala na rozpoznanie przyczyn niezgodności lub nieodpowiednich decyzji. W przykładowej bazie widać odnotowane błędy polegające na wpisaniu nieodpowiedniej ceny (zapis 2 w relacji Cena3), a także niewłaściwego określenia dostawcy towaru.

Czas transakcyjny może poprzedzać rzeczywisty, gdy informację o jakimś fakcie wprowadzono zanim miał on miejsce w rzeczywistości. Czas transakcyjny, późniejszy



Rys.3. Graficzna reprezentacja relacji (bi)temporalnej. Źródło. [7]

Tabela 3. Relacja: Cena3

Towar	Cena	Czas rzeczywisty		Czas transakcyjny	
		Data pocz.	Data końc.	Data pocz.	Data końc.
łyżki	15	01.03.94	25.04.94	02.03.94	nadal
łyżki	18	01.05.94	nadal	01.05.94	04.05.94
łyżki	20	01.05.94	31.05.94	05.05.94	nadal
łyżki	30	01.06.94	nadal	30.05.94	nadal
widelce	30	10.03.94	01.09.94	10.03.94	nadal
noże	40	01.05.94	nadal	03.05.94	nadal

Relacja: Dostawca3

Towar	Dostawca	Czas rzeczywisty		Czas transakcyjny	
		Data pocz.	Data końc.	Data pocz.	Data końc.
łyżki	A	01.03.94	30.04.94	21.02.94	nadal
łyżki	C	20.05.94	nadal	19.05.94	09.06.94
łyżki	B	20.05.94	nadal	10.06.94	nadal
widelce	A	05.05.94	20.05.94	10.05.94	nadal
widelce	B	25.05.94	04.06.94	30.05.94	nadal
widelce	A	25.06.94	nadal	05.06.94	nadal
noże	C	01.05.94	nadal	03.05.94	nadal

niż rzeczywisty, wskazuje na opóźnienia we wprowadzaniu informacji do bazy. Czas transakcyjny nie może jednak być nigdy późniejszy niż czas aktualny. Zapisy czasu transakcyjnego mogą służyć analizie „wąskich gardeł" systemów oraz ujawniać źródła błędnych decyzji, które oparto na zdezaktualizowanych danych.

Bitemporalne bazy danych są również odpowiedzią na problem korekty danych w taki sposób, aby błędna informacja nie została stracona. Podkreślić należy, że w bazach danych tego typu nie ma tradycyjnie rozumianej poprawy błędów. Wszelkie dane raz wpisane do bazy pozostają w niej niezmienione, a wartości poprawne zostają wprowadzone do bazy w formie nowych zapisów. **Błędne zapisy są traktowane jako element historii opisu obiektu.**

Jakie korzyści niesie ze sobą ujęcie czasu w bazie danych z punktu widzenia użytkownika i manipulowania danymi? Może powstać wątpliwość, czy aby gromadzenie tak dużej ilości zapisów rzeczywiście jest potrzebne i uzasadnione. Wiadomo, że w istniejących już SI problem reprezentacji czasu przecież jakoś jest rozwiązywany i użytkownicy raczej nie skarżą się na niedobór informacji. Jednakże właściwym argumentem dla wątpiących w potrzeby konstruowania temporalnych systemów baz danych (w tym temporalnych SZBD) jest udzielenie odpowiedzi na pytanie: jak często użytkownicy nawet nie próbują formułować swych problemów, skutecznie odstraszani kosztami konstrukcji specyficznych procedur pozwalających na odkrycie dynamiki zjawisk posługujących się danymi zgromadzonymi w systemie? Pozostaje problemem otwartym, w jakim stopniu łatwiejsze konstruowanie pytań o dane temporalne wyzwoli potrzebę stosowania rozpatrywanego podejścia do modelowania rzeczywistości.

Języki manipulacji danymi temporalnymi

Jedną z podstawowych korzyści płynących z TBD jest możliwość zastosowania nowych konstrukcji językowych, a tym samym zwolnienie programistów z uciążliwego tworzenia własnych procedur. Projektantowi systemu pozwala to na bardziej jednoznaczne ujęcie dynamiki w projekcie systemu i zaoferowanie modelu bardziej zbliżonego do rzeczywistości.

Opisane typy baz danych umożliwiły stworzenie nowych języków do manipulowania danymi temporalnymi oraz zadawania pytań. Zaproponowano wiele koncepcji teoretycznych, które można podzielić na dwie główne grupy:

1. Języki oparte o istniejące języki zapytań.
2. Nowe konstrukcje stworzone specjalnie dla potrzeb technologii TBD.

Pierwsza grupa języków stanowi rozszerzenie wybranego, istniejącego języka o klauzule, które umożliwiają komunikację z TBD. Do tej grupy zaliczyć można np. Historical Query Language HQUEL [12] i Temporal Query Language TQUEL [8].

Nowe języki obecnie opracowywane są oparte na tradycyjnych zasadach, jak na przykład Legol 2.0, języki graficzne dwu- i trójwymiarowe, gdzie czas reprezentowany jest jako trzeci wymiar tradycyjnej bazy danych, Time-By-Example [13] oraz języki wykorzystujące nowe operatory logiczne np.: czasową równość $=_T$, koniunkcję AND_T, negację NOT_T [4] i inne.

Ze względu na zakres zastosowań wyróżnić można trzy grupy języków:

1. uwzględniające czas rzeczywisty (HSQL, TSQL),
2. uwzględniające czas rzeczywisty i transakcyjny (TSQL2),
3. pozwalające formułować pytania dotyczące dynamiki powiązań i konstrukcji relacji (SQL/SE) [10].

Rezygnując z przedstawienia formalnych podstaw konstrukcji języka, aby zaprezentować nowe możliwości zadawania pytań posłużymy się jednym z proponowanych rozszerzeń języka SQL – językiem TSQL (Temporal SQL) w odniesieniu do danych z tablic CENA2 (CENA) I DOSTAWCA2 (DOST) [5] [6].

Konstrukcja języka wymaga przeprowadzenia normalizacji czasowej. Czas jest więc atrybutem całej relacji, ponieważ jedna relacja może zawierać tylko jeden atrybut, którego wartości zmieniają się w czasie.

1. Deklaracja tablic

Definicja tablic w TSQL jest podobna jak w standardzie SQL, uzupełniona o dodatkowe, specjalnie stworzone atrybuty, zaznaczające, że relacja jest relacją temporalną.

CENA(Towar, Cena, T_S, T_E),
DOST(Towar, Dostawca, T_S, T_E),

gdzie T_S i T_E oznaczają odpowiednio czas początkowy (TIME-START) i końcowy (TIME-END) interwału, w którym wartość atrybutu przyjmowała określoną wartość. Dla oznaczenia tego odstępu czasu wprowadzono operator .INTERVAL. Oczywiście programista jest zwolniony z programowej obsługi czasu, gdyż wszystkie związane z tym czynności przejmuje SZBD.

2. Klauzula WHEN

Klauzula WHEN jest podobna do już stosowanej klauzuli WHERE, z tą różnicą, że określa się w niej zależności dotyczące czasu. W tej klauzuli można posłużyć się predefiniowanymi operatorami czasowymi:

```
[t1, t2] BEFORE [t3, t4] ⇔ t2<t3
[t1, t2] DURING [t3, t4] ⇔ t1≥t3 ∧ t2≤t4 i
[t1, t2] AFTER [t3, t4] ⇔ t1>t4
[t1, t2] EQUIVALENT [t3, t4] ⇔ t1=t3 ∧ t2=t4 i
[t1, t2] ADJACENT [t3, t4] ⇔ (t3-t2) = 1 ∨ (t1-t4) = 1
[t1, t2] OVERLAP [t3, t4] ⇔ t1≤t4 ∧ t3≤t2 i
[t1, t2] FOLLOWS [t3, t4] ⇔ (t1-t4) = 1
[t1, t2] PRECEDES [t3, t4] ⇔ t3-t2 = 1
```

gdzie t1, t2, ... oznaczają odpowiednie chronony.

Użycie tej klauzuli wraz z operatorami pozwala na zadawanie pytań typu:

Jakie były ceny łyżek gdy ich dostawcą była firma A?

```
SELECT cena
FROM CENA
WHERE CENA.towar = DOST.towar
        AND DOST.towar='łyżki'
        AND DOST.dostawca='A'
WHEN CENA.INTERVAL OVERLAP DOST.INTERVAL
```

Jaka firma dostarczała widelce przed firmą, która dostarcza je obecnie?

```
SELECT B.dostawca
FROM DOST A, DOST B
WHERE A.towar = B.towar
        AND A.towar='widelce'
        AND A.TIME-END=nadal
WHEN B.INTERVAL BEFORE A.INTERVAL    itd.
```

3. Klauzula TIME-SLICE

Każde pytanie o dane zmienne w czasie można uściślić przy pomocy klauzuli TIME-SLICE, dzięki której można wskazać, jakiego okresu czasu rzeczywistego dotyczy pytanie. Oprócz jawnego wskazania badanego okresu (MONTH, YEAR, DAY,...), można to również uczynić przy pomocy stałej NOW oznaczającej aktualny czas. Np. gdybyśmy chcieli ograniczyć zasięg pierwszego pytania do ostatnich trzech miesięcy wystarczyłoby do pytania dopisać:

TIME-SLICE month [NOW – 3, NOW]

gdybyśmy zaś chcieli uzyskać informację z lat 1990–1993:

TIME-SLICE year [1990, 1993].

4. Klauzula MOVING WINDOW

Bardzo pożyteczną jest formuła MOVING WINDOW, pozwalająca na przeglądanie bazy w określonym przedziale czasu. Zwłaszcza w procesie analizy szeregów czasowych i diagnozowania, wyszukiwanie takich szczególnych okresów może pomóc w ustaleniu istotnych zmian w działaniu systemu. Przykładowe pytanie może brzmieć:

Wyświetl towary, których dostawcy zmienili się więcej niż 3 razy w okresach trzymiesięcznych.

```
SELECT towar
FROM DOST
MOVING WINDOW 3 months
HAVING COUNT(*)>3
```

5. Klauzule określania kolejności FIRST, SECOND, THIRD, nTH, LAST

Dzięki tym klauzulom można odpowiedzieć na pytania o kolejne zmiany wartości atrybutów np.:

Kto był drugim w kolejności dostawcą łyżek?

```
SELECT SECOND  Dostawca
FROM DOST
WHERE TOWAR = 'łyżki'
```

Przytoczone przykłady są jedynie niewielką częścią zaproponowanych rozszerzeń języka SQL. W literaturze definiowanych jest wiele propozycji i tylko od wiedzy projektanta i programisty zależy wybór odpowiedniego narzędzia.

Zastosowanie odpowiedniego modelu danych nie jest jedynym warunkiem, który musi być spełniony, aby możliwe było użycie nowych konstrukcji językowych. Język manipulacji danymi jest jedynie uzewnętrzeniem możliwości SZBD, tak więc jego rozbudowa jest kolejnym warunkiem wprowadzenia wymiaru czasu do bazy danych.

* * *

Technologia TBD stanowi odpowiedź na wyzwania stojące przed nowoczesnymi SI. Dzięki jej zastosowaniu SI będą mogły lepiej odwzorowywać rzeczywistość i lepiej zaspokajać potrzeby użytkowników. Projektantom i programistom pozwoli szybciej, łatwiej i wydajniej tworzyć SI, w których istotną rolę odgrywa dynamika zmian.

Technologia ta jest powiązana z innymi rozwijającymi się obszarami badań nad bazami danych: wielowersyjnymi BD, które można traktować jako szczególny przypadek TBD baz danych oraz systemami wielobazowymi.

Problem czasu nie jest problemem jedynie systemów bazodanowych. Może on odgrywać ważną rolę również w systemach wnioskowania (w systemach baz wiedzy) oraz w inżynierii programowania. Są one zresztą coraz bardziej ze sobą powiązane, a obecnie coraz mocniej przenikają się wzajemnie.

Autorzy stanęli przed problemem wprowadzenia polskich odpowiedników niektórych, specjalistycznych terminów angielskich. Świadomi niedoskonałości zaproponowanych rozwiązań będą wdzięczni za wszelkie uwagi i propozycje dotyczące polskiej terminologii dla omawianej dziedziny.

LITERATURA

[1] Kania K., Gołuchowski J.: Zagadnienia reprezentacji czasu w klasycznych systemach baz danych. Informatyka 7/1996

[2] Kania K., Gołuchowski J.: Reprezentacja czasu w modelach konceptualnych SI. Informatyka 8/1996

[3] Ling D., Bell D.: Modelling and Managing Time in Database Systems. The Computer Journal, vol. 35, no. 4, 1992

[4] Llewellyn M., Bassiouni M.Historical Database Views.Information and Software Technology, vol. 33, no. 2, 1991

[5] Navathe S., Ahmed R.: TSQL: A Language Interface for History Database. w: Temporal Aspects in Information Systems. Rolland C., Bodart F., Leonard M. (Eds.) North-Holland, Amsterdam, 1988

[6] Navathe S., Ahmed R.: Temporal Extension to the Relational Model and SQL. w: Temporal Databases. The Benjamin/Cummings Publishing Company Inc., 1993

[7] Snodgrass R., Ahn I.Temporal Databases. Computer. no. 19, 1986

[8] Snodgrass R.: The Temporal Query Language TQUEL. ACM TODS 12, 1987

[9] Snodgrass R.: Temporal Databases. Materiały konferencji Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Piza, 1992

[10] Roddick J.F. SQL/SE A Query Language Extension for Databases Supporting Schema Evolution, SIGMOD RECORD, vol. 21, no. 2, 1992

[11] Tansel A.: Modelling Temporal Data. Information and Software Technology, vol. 32, no. 8, 1990

[12] Tansel A.: A Historical Query Language. Information Sciences, no. 53, 1991

[13] Tansel A., Erol Arkun M., Ozsoyoglu G.: Time-by-Example Query Language for Historical Databases. IEEE Transaction on Software Engineering, vol. 15, no. 4, 1989.

Wykorzystanie strukturalnych technik...

dokończenie ze s. 16

Wszystkie diagramy muszą być weryfikowane przez przedstawicieli użytkownika. Weryfikacja ta odbywa się najczęściej w kilku etapach i każda następna konsultacja zbliża w sposób „iteracyjny" do modelu prawidłowo opisującego stan rzeczywisty.

Po etapie weryfikacji diagramów, opisujących stan aktualny, buduje się model systemu docelowego, w którym ponownie w wyniku kilku spotkań z użytkownikiem uzgadnia się wprowadzenie zmian w strukturze organizacyjnej przedsiębiorstwa oraz sposobie przepływu informacji wynikających z wdrożenia systemu powielanego.

Zdecydowaliśmy się na wykorzystanie w pracach analitycznych najnowszych metod gromadzenia i przetwarzania danych. W reprezentowanej przez nas firmie, w związku z rozproszeniem terytorialnym, istnieje kilka działów analiz. Dlatego, w celu ujednolicenia sposobu tworzenia dokumentacji i prezentacji danych, zdecydowaliśmy się na wykorzystywanie tych samych narzędzi, którymi są oprócz EasyCASE: URD (User Requirements Document – specyfikacja wymagań użytkownika), SRS (Software Requirements Specyfication – specyfikacja wymagań na oprogramowanie) oraz analizę SWOT (Strenghts, Weakness, Opportunities, Threathnes).

Zdobyte przez nas doświadczenia dowodzą, że sztukę pracy z EasyCASE opanowuje się bardzo szybko a umiejętność ta w połączeniu z odpowiednim przygotowaniem merytorycznym analityka pozwala mu w znacznej mierze usprawnić proces analizy systemu informacyjnego. Z drugiej strony graficzna prezentacja wyników analizy jest bez większych problemów interpretowana przez osoby, korzystające z jej wyników, i pozwala uniknąć pracochłonnego sporządzania, niezbyt w końcu skutecznych, opisów słownych.

Na koniec należy podkreślić, że pakiet EasyCASE zawiera również inne, nie mniej cenne mechanizmy wspomagające procesy analizy i projektowania systemów informatycznych, których omówienie przekracza ramy niniejszego artykułu.

Paweł Wiechecki i Piotr Kuboszek są pracownikami Działu Analiz Systemów Zarządzania w firmie Computer Systems For Business International S.A.

Szanowni Czytelnicy!

Złośliwy chochlik drukarski działał w numerze 8/96. W artykule „OB! OK! - obiektowy system zarzadzania bazą danych" Pana Macieja Piaseckiego zamiast tabeli nr 1 wstawił jej lustrzane odbicie. Redakcja serdecznie przeprasza czytelników i autora za ten błąd.

Tabela 1. Typy proste oraz typy predefiniowane danych w systemie *OB!OK!*

nazwa typu	opis	nazwa typu	opis
CHAR	typ znakowy (**char**)	**TIME**	predefiniowana klasa *C++* **CDataTime** zapewniająca reprezentację danych typu czas (ale bez daty)
STRING	ciąg znaków (**char***)	**DATE**	predefiniowana klasa *C++* **CDataDate** reprezentująca datę (ale bez czasu)
BYTE	liczba całkowita bez znaku (**unsigned short int**)	**FULLDATE**	predefiniowana klasa *C++* **CDataFullDate** reprezentująca pełne dane o dacie: data i czas
INTEGER	liczby całkowite (**int**)	**GRAPHICS**	predefiniowana klasa *C++* **CDataGraphics** reprezentująca dowolny format danych graficznych o charakterze statycznym (np. zdjęcie)
LONG	liczby całkowite o podwójnym zakresie (**long int**)	**MULTIMEDIA**	predefiniowana klasa *C++* **CDataMultimedia** reprezentująca dowolny format danych multimedialnych, w tym i dane graficzne o charakterze dynamicznym (np. film video)
DOUBLE	liczby zmiennoprzecinkowe o dużej precyzji (**double**)	<typ obiektowy>	dowolna klasa danych zdefiniowana w schemacie bazy
VOID*	dowolny blok danych - ciąg bajtów (analogia do wskaźnika na obszar pamięci)		

Reengineering - a po polsku?

Jerzy Szyller

Ojcowie *„reengineeringu"*, za jakich przez środowisko biznesu uznawani są Michael Hammer i James Champy, w swojej fundamentalnej książce „Reengineering the Corporation" podają zwięzłą definicję tego pojęcia. Według nich *„Reengineering"*, to *„przemyślenie od podstaw i radykalne przeprojektowanie procesów związanych z działalnością gospodarczą* (tj. biznesowych – przyp. J.Sz) *w celu osiągnięcia zdecydowanego polepszenia bieżących, zasadniczych miar wydajności* (przedsiębiorstw, organizacji – przyp. J.Sz), *takich jak koszty, jakość, szybkość działania i poziom obsługi klientów"*. Ta nieco pompatyczna definicja nie pozostawia cienia wątpliwości, że **proces** *„reengineeringu"* powinien przebiegać maksymalnie szybko, docierać do podstaw prowadzonego biznesu, radykalnie go przekształcać, powodując dramatyczną zmianę różnych parametrów, najlepiej wszystkich tych razem, które wpływają na wydajność całej organizacji. Zatem *„reengineering"* proponuje zestaw rewolucyjnych środków i przepisów na ratowanie tych przedsiębiorstw i organizacji, które przestały sobie radzić na współczesnym rynku, staczając się po równi pochyłej ku, łatwemu do przewidzenia, bankructwu.

W Polsce *„reengineering"* dopiero zaczyna się przebijać do świadomości zarządów i kół kierowniczych przedsiębiorstw, które bardzo często uważają, że ten problem ich nie dotyczy. Mimo to coraz częściej czytamy, że znajdują się odważni, którzy nie mając wiele do stracenia decydują się na dokonanie rewolucyjnych zmian w przedsiębiorstwie. Właśnie na *„reengineering"*. Ponieważ pojęcie to będzie stosowane coraz częściej, warto zastanowić się, czy znowu przejmiemy nazwę angielską, czy pokusimy się o sprawne spolszczenie terminu. Tym razem sprawa nie wygląda beznadziejnie.

Przytoczonej definicji odpowiada w języku polskim kilka mniej lub bardziej precyzyjnych pojęć, używanych przez różnych autorów – rekonstrukcja, reorganizacja, restrukturyzacja. Ostatnio pojawiła się też propozycja – reinżynieria (zarządzania). Ponieważ chodzi o osiągnięcie poprawności językowej przyjętego terminu w języku polskim, dalej przytoczę znaczenia proponowanych polskich odpowiedników, które znalazłem w słowniku języka polskiego pod redakcją profesora Witolda Doroszewskiego.

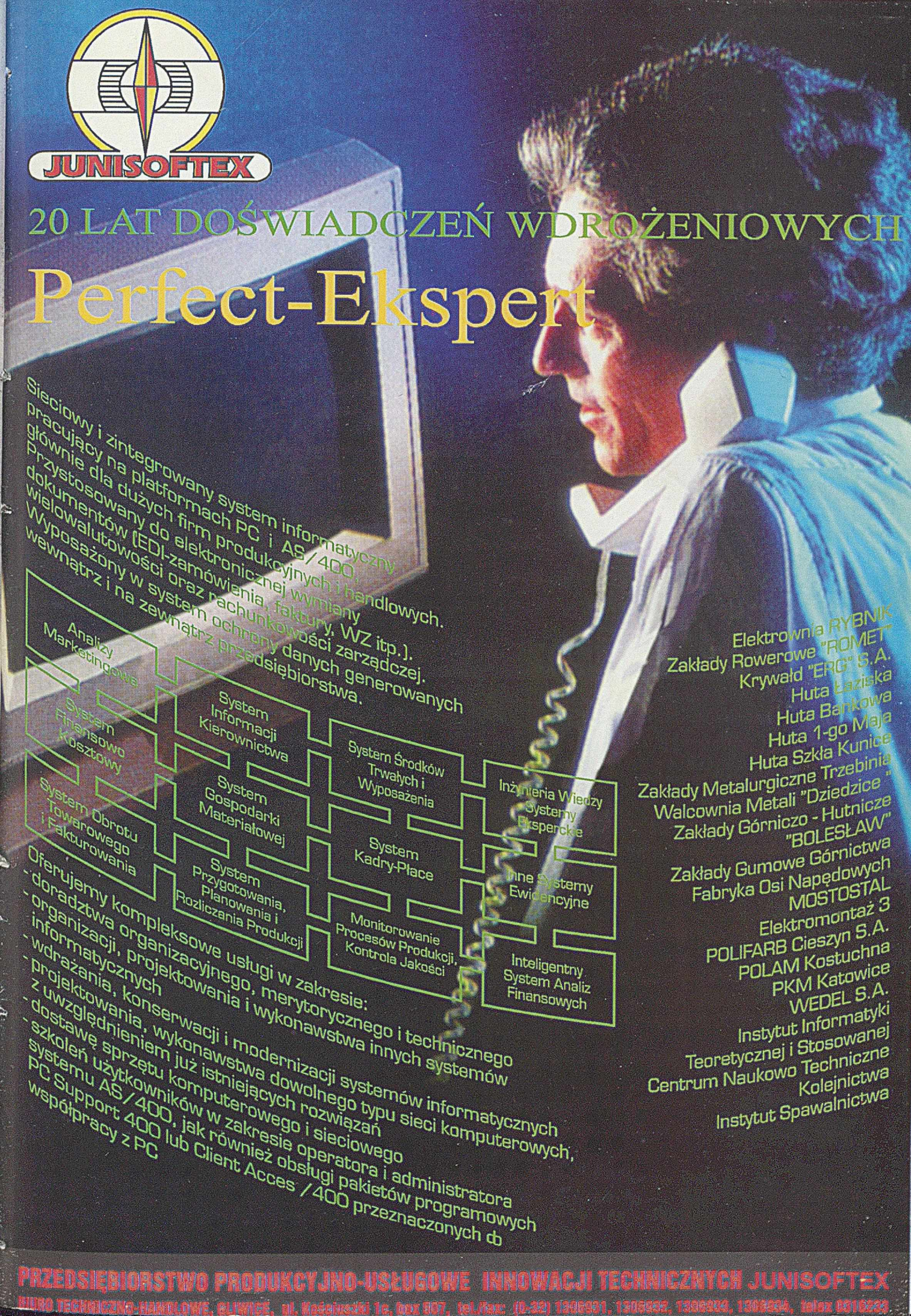
Rekonstrukcja – *odtworzenie czego na podstawie dochowanych fragmentów, szczątków, przekazów, form pochodnych, itp., zwłaszcza: odbudowanie, odtworzenie na takich podstawach zniszczonych, częściowo lub całkowicie, budynków albo dzieł sztuki.* W procesie *„reengineeringu"* zdecydowanie nie chodzi o odtwarzanie w dawnej postaci. Wręcz przeciwnie – mamy tutaj do czynienia z budowaniem nowej jakości.

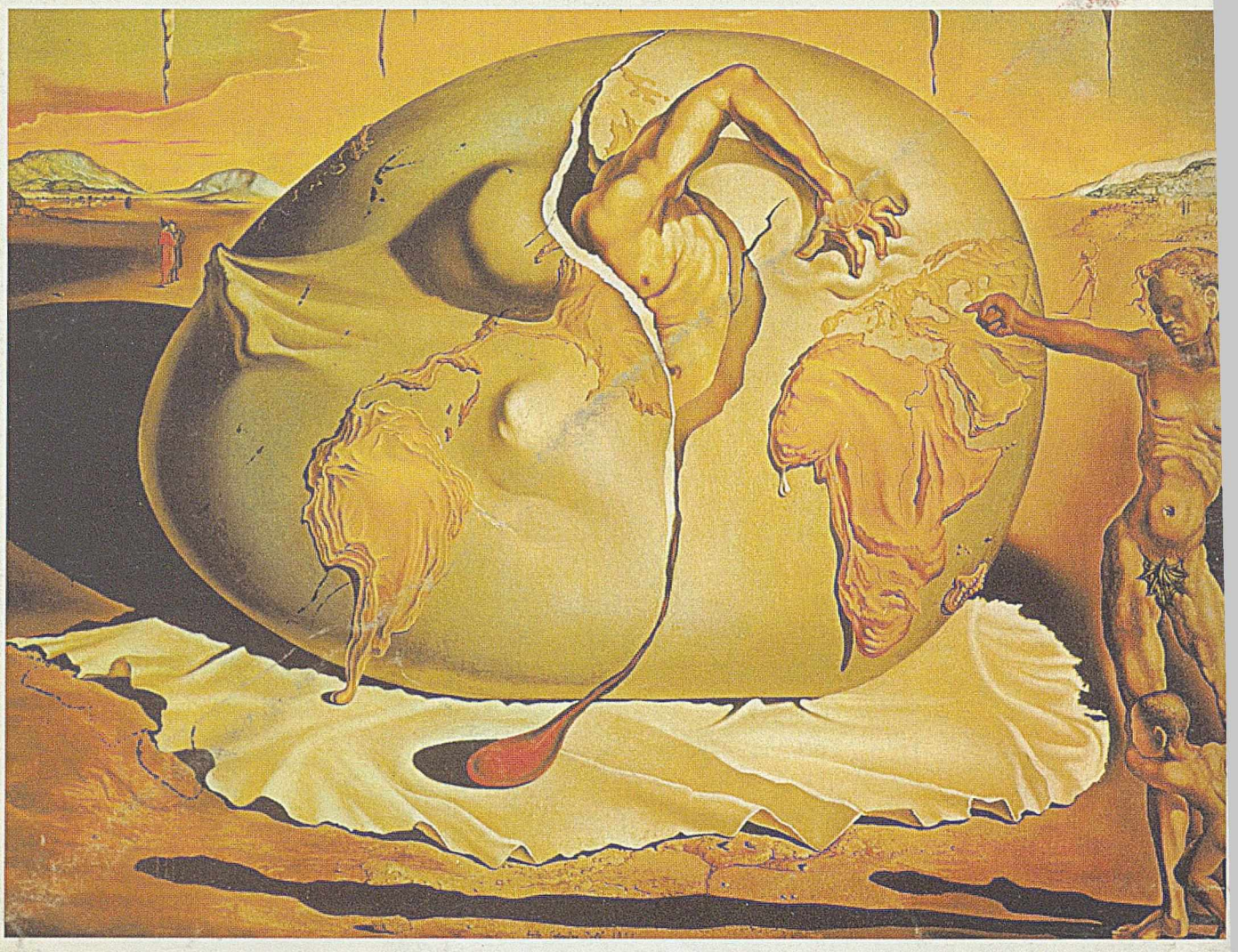
Reinżynieria – niestety takiego słowa w języku polskim nie ma. To oczywiście nie przeszkadza, aby je utworzyć. Zatem czym jest inżynieria? **Inżynieria** – *nauka wykonywania robót inżynierskich, dział uczelni technicznej kształcący inżynierów*. Należy jeszcze dodać, że w informatyce bardzo ważną dziedziną jest **inżynieria oprogramowania**, skupiająca się na badaniu metod i środków produkcji oprogramowania. Jednakże znaczenie angielskiego słowa *„engineering"*, w którym kryje się również proces twórczej inwencji, jest znacznie głębsze i subtelniejsze od polskiego słowa *„inżynieria"*, które przez długie lata wiązało się z działalnością raczej wojskową. Osobiście uważam, że inżynieria, w odniesieniu do spraw ekonomicznych i społecznych, zwłaszcza w minionym systemie komunistycznym, zwykle kojarzyła się fatalnie z manipulacją. Nie włączajmy więc inżynierów zbyt daleko w proces naprawy przedsiębiorstw i organizacji.

Restrukturyzacja – niespodzianka, to słowo nowe, którego nie ma we wspomnianym słowniku języka polskiego (sic!). Natomiast znajduje się w słowniku wyrazów obcych z 1993 roku. Hasło brzmi – *przywrócenie dawnej struktury, postaci czegoś; odbudowanie*. Mamy więc problem, ponieważ według ducha języka polskiego znowu wracamy, wbrew intencji pomysłodawców *„reengineeringu"*, do pierwotnej postaci struktury, procesów, konstrukcji itp. Coś tu się jednak nie zgadza. Restrukturyzację, zwłaszcza ostatnio, zazwyczaj kojarzymy ze zmianami, a nie konserwowaniem przeszłości. To słowo, które przecież zrobiło olbrzymią karierę w ciągu ostatnich lat w kontekście polskiej gospodarki. Zrestrukturyzowaliśmy, ponieważ będzie on działać według całkiem nowych zasad, prawie cały handel, zaczynamy restrukturyzować administrację, następny jest przemysł. Tu widzimy przykład szybszego rozwoju języka potocznego niż słownikowego.

Reorganizacja – *zmiana organizacji czego, zorganizowanie czego na nowo, w inny sposób*. To dobre określenie procesu *„reengineeringu"*. Definicja w języku polskim jest zbliżona do angielskiej. Sądzę, że właściwego wyboru pomiędzy **restrukturyzacją** a **reorganizacją** można dokonać na podstawie analizy słów **struktura** i **organizacja**. **Organizacja** – *struktura, budowa (substancji, organizmu); konstytucja fizyczna lub psychiczna, natura, usposobienie*. **Struktura** – *układ i wzajemne relacje elementów stanowiących całość; budowa, ustrój lub całość zbudowana w pewien sposób z poszczególnych elementów; zespół*. Łatwo zauważyć, że pojęcie **struktury** jest znacznie pojemniejsze treściowo, zawierając w sobie również element organizacji. Przecież struktura z definicji musi być zorganizowana. *„Reengineering"* obejmuje przede wszystkim zmianę procesów biznesowych, a więc właśnie to, co zachodzi na poziomie wzajemnych relacji pomiędzy elementami składającymi się na całość organizacji.

Tak więc, mimo pewnych niezgodności semantycznych z definicją słownikową zdecydowanie skłaniam się do tłumaczenia *„reengineeringu"* jako **restrukturyzacji**. Nie chcąc jednak aspirować do roli językoznawcy, którym nie jestem, proponuję wszystkim, którzy będą mieli inne propozycje udostępnić nasze łamy. Redakcja „Informatyki" zaprasza do dyskusji także nad innymi kontrowersyjnymi pojęciami i terminami, które pojawiają się na łamach prasy informatycznej i przysparzają kłopotów z interpretacją i tłumaczeniem.


JUNISOFTEX
20 LAT DOŚWIADCZEŃ WDROŻENIOWYCH
Perfect-Ekspert
Sieciowy i zintegrowany system informatyczny pracujący na platformach PC i AS/400, głównie dla dużych firm produkcyjnych i handlowych. Przystosowany do elektronicznej wymiany dokumentów (EDI-zamówienia, faktury, WZ itp.). Wielowalutowości oraz rachunkowości zarządczej. Wyposażony w system ochrony danych generowanych wewnątrz i na zewnątrz przedsiębiorstwa.
Analizy Marketingowe
System Finansowo Kosztowy
System Obrotu Towarowego i Fakturowania
System Informacji Kierownictwa
System Gospodarki Materiałowej
System Przygotowania, Planowania i Rozliczania Produkcji
System Środków Trwałych i Wyposażenia
System Kadry-Płace
Monitorowanie Procesów Produkcji, Kontrola Jakości
Inżynieria Wiedzy Systemy Eksperckie
Inne Systemy Ewidencyjne
Inteligentny System Analiz Finansowych
Oferujemy kompleksowe usługi w zakresie:
- doradztwa organizacyjnego, merytorycznego i technicznego
- organizacji, projektowania i wykonawstwa innych systemów informatycznych
- wdrażania, konserwacji i modernizacji systemów informatycznych
- projektowania, wykonawstwa dowolnego typu sieci komputerowych, z uwzględnieniem już istniejących rozwiązań
- dostawę sprzętu komputerowego i sieciowego
- szkoleń użytkowników w zakresie operatora i administratora systemu AS/400, jak również obsługi pakietów programowych PC Support 400 lub Client Acces /400 przeznaczonych do współpracy z PC
Elektrownia RYBNIK
Zakłady Rowerowe "ROMET"
Krywałd "ERG" S.A.
Huta Łaziska
Huta Bankowa
Huta 1-go Maja
Huta Szkła Kunice
Zakłady Metalurgiczne Trzebinia
Walcownia Metali "Dziedzice"
Zakłady Górniczo - Hutnicze "BOLESŁAW"
Zakłady Gumowe Górnictwa
Fabryka Osi Napędowych MOSTOSTAL
Elektromontaż 3
POLIFARB Cieszyn S.A.
POLAM Kostuchna
PKM Katowice
WEDEL S.A.
Instytut Informatyki Teoretycznej i Stosowanej
Centrum Naukowo Techniczne Kolejnictwa
Instytut Spawalnictwa
PRZEDSIĘBIORSTWO PRODUKCYJNO-USŁUGOWE INNOWACJI TECHNICZNYCH JUNISOFTEX
BIURO TECHNICZNO-HANDLOWE, KLUCZE, ul. Kościuszki 1a, box 807, tel./fax: (0-32) 1306931, 1306932, 1306933, 1306934, telex 0316931



BUSINESS PARTNER

digital

Microsoft®
SOLUTION PROVIDER
Authorized Technical Education Center

Z ogromną przyjemnością zawiadamiamy Państwa, iż z dniem 01.09.1996 rozpoczęło działalność Centrum Szkoleniowe HECTOR będące kontynuatorem pracy Centrum Szkoleniowego DIGITAL w Polsce. Centrum Szkoleniowe HECTOR prowadzi kursy w oparciu o oryginalne materiały szkoleniowe DIGITAL-a, obejmujące wykłady i ćwiczenia z indywidualnym dostępem do najnowocześniejszego sprzętu w czasie trwania zajęć. Oferujemy cykle szkoleń zróżnicowane pod względem poziomu, dostosowane do indywidualnych potrzeb klientów i prowadzone przez najlepszych specjalistów. Zakres kursów obejmuje m. in.: systemy operacyjne WINDOWS NT, DIGITAL UNIX, OPEN VMS, bazy danych, sieci komputerowe, platformy sprzętowe ALPHA i in. Oferujemy szkolenia specjalistyczne na indywidualne zamówienia firm.

Nasze Centrum Szkoleniowe jest jedynym w Polsce partnerem międzynarodowej organizacji edukacyjnej Global Knowledge Network posiadającej swoje filie w 44 krajach Świata, zatrudniającej ponad 700 instruktorów, którzy do chwili obecnej wyszkolili już ponad 100 000 osób. Jest to jedyne centrum szkoleniowe produktów DIGITAL-a /programowych i sprzętowych/ oraz największe centrum Solution Provider MICROSOFT na Świecie. Współpracując z Global Knowledge Network organizujemy dowolne kursy w kraju, a w przypadku jednostkowych zamówień na specjalistyczne szkolenia oferujemy takie usługi za granicą.

Szczegółowe informacje, harmonogram, cennik, programy dostępne są w siedzibie naszej firmy. Serdecznie zapraszamy !

hector S.A

Authorized Training Center

GlobalKnowledgeNetwork
CONNECTING PEOPLE TO KNOWLEDGE

• ul. Gwiaździsta 19 • 01-651 Warszawa • tel. +48 /22/ 330 285 , 333 992 , 333 993 • fax +48 /22/ 333 964 •
• komertel 3912 3065 • bbs +48 /22/ 337 258 • Internet : http://www.atm.com.pl/~hector •